

Лукашенко В.В., аспирант  
Рязанский государственный университет имени С.А. Есенина

## РАЗРАБОТКА МЕТОДА РЕШЕНИЯ ЗАДАЧИ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЫ КЛАСТЕРА НА НЕЙРОКОМПЬЮТЕРАХ

v.lukashenko@nstlabs.net

В статье предлагается решение задачи поиска элементарных фрагментов программ, и построения итоговой вычислительной структуры кластера. Метод представлен с позиции теории искусственных нейронных сетей. Фрагменты задач представляются во внутренней форме представления программ – постфиксной Польской форме. В зависимости от того, каким образом фрагментам требуются данные, выстраивается итоговая вычислительная структура кластера одного из типов: векторная, конвейерная, векторно-конвейерная, конвейерно-векторная, матричная.

**Ключевые слова:** кластеризация, нейрокомпьютерные технологии, распределенные вычисления, нейровычисления, распределенные нейровычисления.

В современном научном и производственном мире остро стоят проблемы существенной недостаточности вычислительных ресурсов для решения ряда задач. Устоявшимся решением данной проблемы является использование параллельных или распределенных вычислений.

В основе любых систем распределенных вычислений лежит модель вычислений. Модель вычислений связывает между собой архитектуру и модель программирования в распределенных системах, а также отражает взаимодействие процессов. Но на текущее время для классифицированных технологий распределенных вычислений универсальных моделей не построено. Это объясняется тем, что модели вычислений крайне зависимы от выбора архитектуры распределенной системы, и при выборе нестандартной архитектуры для решения специфических задач необходимо также решать задачу построения модели вычислений.

В работе [1] был предложен алгоритм выбора структуры обработки информации, который является неотъемлемой частью модели вычислений распределенного вычислительного кластера с нейрокомпьютерной архитектурой. В нейрокомпьютерных системах обработка информации производится в специализированных вычислительных структурах: векторных, конвейерных, векторно-конвейерных, конвейерно-векторных, матричных [2]. Нами был предложен вариант перенесения принципа обработки информации с нейрокомпьютерной системы на нейрокластерную.

Основной задачей данной статьи является поиск элементарных фрагментов программ, и построения итоговой вычислительной структуры кластера на нейрокомпьютерах. Решение этой задачи осложнено неопределенностью условий. Так, например, в одном и том же отрезке программного кода, выделенного в результате разбиения задач на подзадачи, данные

могут требоваться лишь части переменных на разных итерациях вычисления.

Задачи с подобными неопределенными условиями рассматриваются и с успехом решаются в теории нейронных сетей [2]. Алгоритм выбора вычислительной структуры распределенного кластера примет вид искусственной нейронной сети.

Для разработки искусственной нейронной сети необходимо рассмотреть исчерпывающее множество событий. *Исчерпывающее множество событий (ИМС)* – совокупность событий, высказывания о которых покрывает весь возможный смысловой диапазон проявления объекта высказывания. Каждая допустимая ситуация характеризуется тем, что значение ИСТИНА – 1 может принимать единственное высказывание из этой совокупности, значение 0 могут принимать все высказывания [2].

Исчерпывающее множество событий рассматриваемой задачи состоит из:

- числа классов эквивалентности  $L$ .
- Порядков классов  $|a_i| = 1$ .
- Ситуаций, когда данные между фрагментами в классах передаются последовательно.
- Ситуаций, когда данные передаются одновременно для всех фрагментов классов.
- Ситуаций, когда данные между фрагментами в классах передаются последовательно с редким исключением фрагментов  $l$ -го класса.
- Ситуаций, когда данные между фрагментами в классах передаются последовательно, кроме  $l$ -го, а  $l$ -му классу, в свою очередь, данные передаются одновременно.
- Ситуаций, когда данные передаются одновременно для всех фрагментов классов, данные между фрагментами  $l$ -го класса передаются последовательно.

Различные комбинации этих утверждений образуют ИМС.

Дадим каждому утверждению условное обозначение:  $A, B_i, i = \overline{1, L}, C_1, C_2, C_3, C_4, C_5$  по порядку соответственно.

Таким образом, искусственная нейронная сеть примет вид (рис. 1), где векторная, конвей-

ерная, векторно-конвейерная, конвейерно-векторная, матричная – вычислительные структуры распределенного вычислительного кластера.

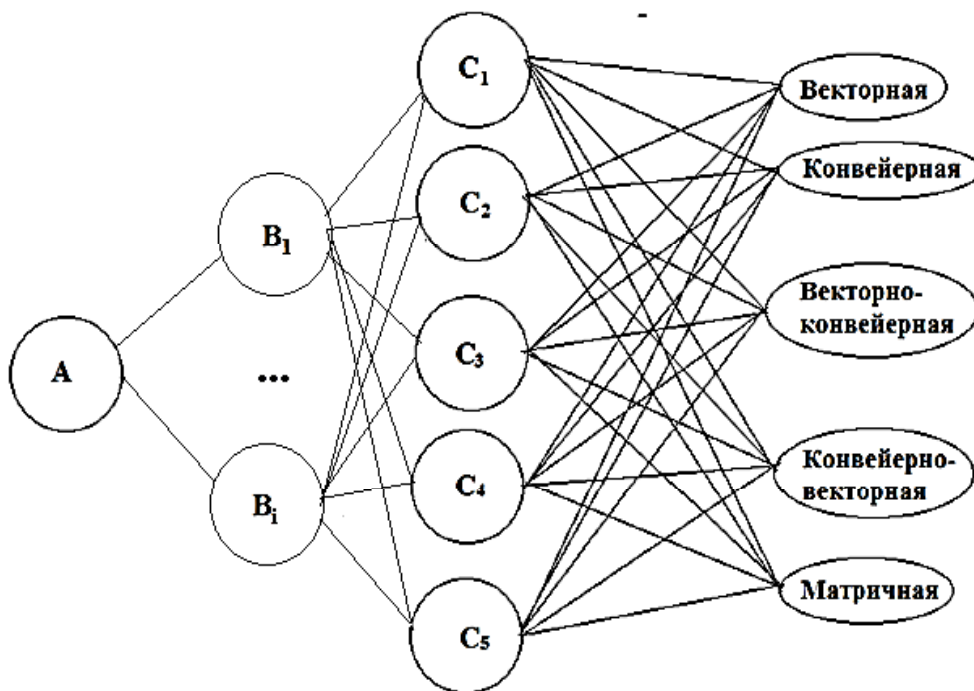


Рис. 1. Схема искусственной нейронной сети, описывающая все ИМС

Схема, представленная на рисунке 1, описывает все исчерпывающее множество событий. Следующим шагом станет обучение этой сети эталонным набором утверждений, которые справедливы для описания всех вариаций существующих разбиений алгоритмов:

1. Если число классов эквивалентности равно  $L$ , и порядок каждого  $|a_i| = 1$ , и

- обмен данными между фрагментами в классах осуществляется последовательно, - то «конвейерная структура»;

- информация требуется одновременно всем фрагментам классов, то – «векторная структура» [2].

2. Если число классов эквивалентности  $L$  равно единице и порядок каждого класса  $|a_i|$  равен  $l$ , и

- обмен данными между фрагментами в классах осуществляется последовательно, то – «конвейерная структура»;

- информация требуется одновременно всем фрагментам классов, то – «векторная структура» [2].

3. Если число классов эквивалентности  $L$ , и порядок каждого  $|a_i| = 1$ , и

- обмен данными между фрагментами в классах осуществляется последовательно, то – «конвейерная структура»;

- обмен данными между фрагментами в классах осуществляется последовательно с небольшим исключением фрагментов  $l$ -го класса, то – «конвейерная структура»;

- обмен данными между фрагментами в классах, кроме  $l$ -го, осуществляется последовательно, а  $l$ -му классу информация требуется одновременно, то – «конвейерно-векторная структура»;

- фрагменты обмениваются информацией последовательно, однако в некоторых классах информация требуется одновременно, то – «конвейерно-векторная структура» [2].

4. Если порядок каждого класса равен числу классов эквивалентности  $L$ , то – «матричная структура»;

- информация сразу требуется фрагментам всех классов, но в некоторых классах требуется последовательный обмен, то – «векторно-конвейерная структура»;

- информация требуется одновременно фрагментам всех классов, но фрагменты  $l$ -го класса обмениваются последовательно, то – «векторно-конвейерная структура»;

- информация требуется одновременно фрагментам всех классов, кроме некоторых фрагментов  $l$ -го класса, то – «векторно-конвейерная структура»;

- информация требуется одновременно всем фрагментам классов, то – «векторная структура»[2].

Затем необходимо подобрать функцию активации нейронов для каждого слоя, которая адекватно поможет обучить сеть подбором весов.

Из всего многообразия функций активации нейронов каждого слоя, наиболее подходящей является пороговая функция активации.

$$V = \sum_j \omega_j V_j \quad (1)$$

$$V = \begin{cases} V, & V > h \\ 0 & \end{cases}$$

где  $\omega_j$  – синаптический вес входа или вес связи, по которой передается возбуждение от нейрона к нейрону.

Следующим шагом после выбора функции активации нейронов станет обучение нейронной сети методом трассировки. Задачей обучения является выбор единичных весов некоторых связей и нулевых весов других связей так, чтобы в соответствии с функциональным назначением сети сложились отношения «посылка — следствие», подобные тем, которые были созданы выше при построении ИМС. Для достижения требуемого результата необходимо использовать матрицы следования, отображающие как взаимодействие нейронов, так и веса синоптических связей.

Матрица следования для ИМС примет вид (2), где  $V, K, VK, KV, M$  – специально введенные условные обозначения для вычислительных структур: векторной, конвейерной, векторно-конвейерной, конвейерно-векторной и матричной соответственно.

$A$	_	0	0	0	0	0	0	0	0	0	0	0	0	0
$B_1$	0	_	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	_	...	...	...	...	...	...	...	...	...	...	...
$B_i$	0	0	0	_	0	0	0	0	0	0	0	0	0	0
$C_1$	0	0	0	0	_	0	0	0	0	0	0	0	0	0
$C_2$	0	0	0	0	0	_	0	0	0	0	0	0	0	0
$C_3$	0	0	0	0	0	0	_	0	0	0	0	0	0	0
$C_4$	0	0	0	0	0	0	0	_	0	0	0	0	0	0
$C_5$	0	0	0	0	0	0	0	0	_	0	0	0	0	0
$V$	0	0	0	0	0	0	0	0	0	_	0	0	0	0
$K$	0	0	0	0	0	0	0	0	0	0	_	0	0	0
$VK$	0	0	0	0	0	0	0	0	0	0	0	_	0	0
$KV$	0	0	0	0	0	0	0	0	0	0	0	0	_	0
$M$	0	0	0	0	0	0	0	0	0	0	0	0	0	_

После обучения – подбора весовых коэффициентов матрица следования для минимизированной ИМС примет вид. (3).

$A$	_	0	0	0	0	0	0	0	0	0	0	0	0	0
$B_1$	1	_	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	_	...	...	...	...	...	...	...	...	...	...	...
$B_i$	1	0	0	_	0	0	0	0	0	0	0	0	0	0
$C_1$	0	1	0	1	_	0	0	0	0	0	0	0	0	0
$C_2$	0	1	0	1	0	_	0	0	0	0	0	0	0	0
$C_3$	0	1	0	0	0	0	_	0	0	0	0	0	0	0
$C_4$	0	1	0	0	0	0	0	_	0	0	0	0	0	0
$C_5$	0	0	0	1	0	0	0	0	_	0	0	0	0	0
$V$	0	0	0	0	1	1	1	0	0	_	0	0	0	0
$K$	0	0	0	0	1	1	1	0	0	0	_	0	0	0
$VK$	0	0	0	0	0	0	0	1	0	0	0	_	0	0
$KV$	0	0	0	0	0	0	1	1	0	0	0	0	_	0
$M$	0	0	0	0	1	1	0	0	0	0	0	0	0	_

Таким образом, в результате обучения нейронной сети для каждого случая значение функции (1) получается максимальным, из чего можно предположить, что обучение прошло успешно.

Прежде чем начать тестирование работы полученной нейронной сети следует отметить, что для достижения универсальности обработки результатов, необходимо представить их в универсальной форме. Все задачи поставляются в

вычислительный кластер в виде программного кода, а для его представления необходим транслятор для перевода его в машинно-понятную форму, что говорит о жесткой привязке к языку программного кода. Одной из распространённых универсальных форм представления программ является постфиксная польская форма представления программ. Для представления программных конструкций типа цикл, условие и т.д., разработана расширенная постфиксная польская форма [3,6]. Именно в такой промежуточной форме и подается на вход искусственной нейронной сети исходный код программы. В работе используется стандартный алгоритм приведения программного кода в постфиксную Польскую форму. На вход подается программный код, а на выходе получается очередь из операторов и операндов. Полученная очередь подается на вход искусственной нейронной сети.

Опробуем нейронную сеть на реальном примере.

Пример 1.

Пусть есть отрывок исходного программного кода:

```
A=1;
B=2;
C=3;
D=C;
X=A+B;
Y=C+D;
Return x+y;
```

В обратной польской записи выражение примет вид:

$A1=B2=C3=DC=AB+X=CD+Y=XY+\$BR$ .

Деление на минимальные атомарные операции приведет к следующему виду:

1. A1=
2. B2=
3. C3=
4. DC=
5. AB+
6. X=
7. CD+
8. Y=
9. XY+
10. \$BR

Безусловно, можно сократить некоторые бессмысленные операции, в частности операции присваивания и возвращения результата.

1. A1=
2. B2=
3. C3=
4. DC=

5. AB+X=

6. CD+Y=

7. XY+\$BR

Рассуждая по приведенному выше алгоритму получаем, что:

1. Число классов эквивалентности равно единице, и порядок каждого класса равен единице, а информация передается независимо (т.е. параллельно) для случаев A1=, B2=, C3=.

2. Случай DC= выпадает из данного рассуждения, поскольку он не может выполняться параллельно со всеми, т.к. пока не будет инициализировано значение C, не будет и инициализировано значение D. Таким образом DC= информация требуется последовательно. По первым результатам видно, что образуется векторно-конвейерная структура.

3. Далее случаи AB+X= и CD+Y= не могут быть инициализированы до тех пор пока не инициализированы все переменные используемые в этих выражениях. Значит они выполняются последовательно, но относительно друг друга они могут выполняться параллельно.

4. И последнее выражение XY+\$BR выполняется последовательно после инициализации переменных X и Y.

Таким образом для каждого случая получаем максимально значение функции  $V = \sum_j \omega_j V_j$ :

1.  $V = 3, \omega_i = 1$  Векторная
2.  $V = 3, \omega_i = 1$  Векторная
3.  $V = 3, \omega_i = 1$  Векторная
4.  $V = 5, \omega_i = 1$  Конвейерная
5.  $V = 5, \omega_i = 1$  Конвейерная
6.  $V = 5, \omega_i = 1$  Конвейерная
7.  $V = 5, \omega_i = 1$  Конвейерная

A в целом конвейерно-векторная, как показано на рис. 4.

Таким образом, в статье рассмотрен метод поиска элементарных фрагментов программ, и построения итоговой вычислительной структуры кластера на нейрокомпьютерах. Метод представлен с позиции теории искусственных нейронных сетей. Фрагменты задач представляются во внутренней форме представления программ – постфиксной Польской форме. В зависимости от того, каким образом фрагментам требуются данные, выстраивается итоговая вычислительная структура кластера одного из типов: векторная, конвейерная, векторно-конвейерная, конвейерно-векторная, матричная.

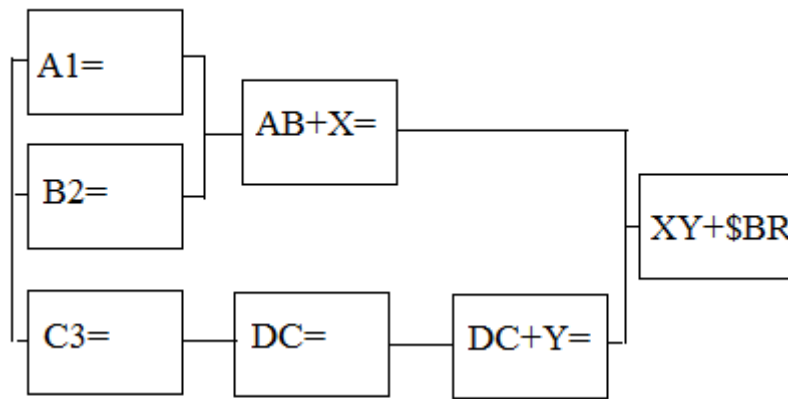


Рис. 4. Результат работы нейронной сети - вычислительная структура для данного отрезка кода

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ручкин В.Н., Романчук В.А., Фулин В.А., Лукашенко В.В. Разработка алгоритма выбора вычислительной структуры распределенного кластера с нейрокомпьютерной архитектурой // Известия Тульского государственного университета. Технические науки. 2015. № 9. С. 236–244.
2. Галушкин А.И. Нейронные сети: основы теории М.: «Горячая линия Телеком», 2010.
3. Топорков В.В. Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.
4. Лукашенко В.В. Анализ основных вопросов классификаций распределенных вычислительных систем // Современная техника и технологии. 2015. № 4 (44). С. 65–69.
5. Бурцев В. С. Параллелизм вычислительных процессов и развитие архитектуры супер ЭВМ. М.: ИВВС РАН, 1997. 152 с.
6. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
7. Корнеев В. В. Параллельные вычислительные системы. М.: Нолидж, 1999. 320 с.
8. Лацис А. Как построить и использовать суперкомпьютер. М.: Бестселлер, 2003. 240 с.
9. Таненбаум Э. Архитектура компьютера. СПб.: Питер, 2002. 704 с.
10. Buyya Ed. R. High performance cluster computing // Architectures and systems. Programming and applications. New Jersey: PrenticeHall PTR, 1999. 354 с.

**Lukashenko V.V.**

### DEVELOPMENT OF METHOD OF CONSTRUCTION OF COMPUTER SOLUTION OF CLUSTER STRUCTURE FOR NEURO-COMPUTERS

*The paper proposes a solution to the elementary fragment program scan tasks, and build the final structure of the computing cluster. The method is presented from the perspective of artificial neural network theory. Fragments of problems presented in the internal form of program presentation - Polish postfix form. Depending on how the fragments of data are required, the total computing cluster built the structure of one of the types: vector, pipelining, vector-conveyor, conveyor-vector, matrix.*

**Key words:** clustering, neurocomputing technology, distributed computing, neurocomputational distributed neurocomputational.

**Лукашенко Владислав Владиславович**, аспирант.  
Рязанский государственный университет имени С.А. Есенина.  
Адрес: Россия, 390000, Рязань, ул. Свободы, д. 46.  
E-mail: v.lukashenko@nstlabs.net