

## ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ КЛАСТЕРА НЕЙРОКОМПЬЮТЕРОВ ДЛЯ ЭМУЛЯЦИИ ТОПОЛОГИИ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

**Лукашенко Владислав Владиславович**

научный сотрудник кафедры ИВТ и МПИ Рязанский государственный университет имени С.А. Есенина, РФ, г. Рязань

**Романчук Виталий Александрович**

кандидат технических наук, доцент кафедры ИВТ и МПИ Рязанский государственный университет имени С.А. Есенина, РФ, г. Рязань

### Parallel programming of the neurocomputer cluster for emulation of the computer network topology

*Lukashenko Vladislav*

*Scientific employee of the department of ICT and MPI Ryazan State University named after S.A. Yesenin, Russia, Ryazan*

**Vitaly Romanchuk**

*Candidate of Technical Sciences, Associate Professor of the Department of ICT and MPI, Ryazan State University named after S.A. Yesenin, Russia, Ryazan*

**Аннотация.** В статье рассмотрен способ программного виртуального эмулирования кластерной топологии более высокого уровня, а затем, и динамического резервирования в полученной топологии вычислительных машин для создания вычислительной нейрокластерной структуры для нужд параллельного программирования.

**Abstract.** The article describes a method for software virtual emulation of a cluster topology of a higher level, and then, and dynamic reservation in the resulting topology of computers to create a computational neural cluster structure for the needs of parallel programming

**Ключевые слова:** распределенной вычисления; кластерные вычисления; кластеризация вычислительных ресурсов; нейровычисления, нейрокомпьютеры; кластеризация нейрокомпьютеров.

**Keywords:** distributed computing; cluster computing; clustering of computing resources; neuro computing; neurocomputers; clustering of neurocomputers.

## Введение

В современной научной и производственной сфере достаточно актуальна задача использования систем распределенной обработки данных из-за недостаточности вычислительных ресурсов для решения ряда задач. Эффективнее всего для решения поставленной проблемы применимы масштабируемые параллельные высокоскоростные кластеры на базе нейрокомпьютеров [3]. В работе рассматривается конкретная реализация вычислительного кластера на базе нейрокомпьютеров MB77.07 ЗАО «НТЦ Модуль», технические характеристики, преимущества и перспективность которых описана в работах [2-8].

Одним из важнейших компонентов любого кластера при его проектировании является топология внутренней вычислительной сети кластера [9]. С другой стороны в нейрокомпьютерных системах обработка поступающих задач происходит посредством специализированных вычислительных структур [2]. В обоих случаях встречаются нестандартные топологии вычислительных сетей связи. В статье рассмотрен способ программного виртуального эмулирования кластерной топологии более высокого уровня, а затем, и динамического резервирования в полученной топологии машин для создания вычислительной структуры. Данный процесс необходим, поскольку в основе идеи разработки кластера на нейрокомпьютерах лежит принцип двойного распараллеливания задач, поступающих на выполнение в такой кластер. Первый этап – это стандартное разбиение задач на атомарные подзадачи, второй этап – размещение на нейрокомпьютеры кластера такого количества задач и данных для них, чтобы их выполнение не занимало более одного нейропроцессорного такта.

Рассмотрим с алгоритмической точки зрения процесс организации и оптимизации четырехмерного гиперкуба на базе стандартной Ethernet сети: выбор первой вычислительной машины кластерной подсети, у которой четвертая секция IP-адреса равна "1", направление broadcast-запроса на широковещательный адрес вычислительной подсети, рекурсивный просмотр tcpdump-файлов каждой вычислительной машины, поиск четырех наименьших значений времени приема пакета, запись в матрицу смежности в соответствующую ячейку последней четвертой секции IP-адреса, переход на шаг 1, выбирая вычислительную машину со следующим порядковым номером 4 секции IP-адреса.

Пример. Пусть есть вычислительный кластер  $\overline{K}$  с количеством вычислительных машин

$\overline{p_i}, i = 1, 10$ . Каждой вычислительной машине назначен IP адрес вида 192.168.0.1-10 и широковещательный адрес 192.168.0.255. Выберем вычислительную машину, у которой 4 секция IP-адреса равна "1". Направляем запрос на широковещательный адрес, для этого воспользуемся специально подготовленной функцией C++.

Формируем очередную ячейку матрицы смежности связей вычислительной подсети кластера.

Следующим шагом исследования является динамическое построение вычислительной нейрокластерной структуры из топологии четырехмерного куба. Рассмотрим модель расписания и критерии по которым составляется расписание планировщиком ресурсов.

Пусть для множества операций  $Op = \langle Op_1, Op_2, \dots, Op_p, \dots, Op_P \rangle$ , связанных отношением последовательности выполнения операций  $Pos$  задан отрезок планирования  $[0, T], t \in [0, T]$  - переменная времени. Пусть  $s_i, f_i$  - моменты начала и окончания операции  $\overline{Op_i}, i = 1, N$ . Тогда расписание в общем случае примет вид:

$$Sh = \{(s_i, f_i), i = \overline{1, N}, s_i, f_i \in [0, T]\} \quad (1)$$

Рассмотрим критерии, влияющие на решение задачи (1). Следует отметить, что первым и самым важным критерием являются количество ресурсов. Чаще всего ресурсы состоят из набора процессоров, памяти и дополнительных ресурсов. Задаются общие уровни наличия каждого типа ресурса во времени:

$$r_t = \{r_t^j, j = \overline{1, J}\}, t \in [0, T], \quad (2)$$

где  $J$  - число типов ресурсов. Причем  $r_t^j$  - не зависит от времени, верхняя граница  $J$ -го типа (число процессоров, пропускную способность коммуникационной системы) будем обозначать  $r_0^j$ .

Следующим критерием является система операций. Система операций описывается следующим образом: задается множество операций, которые необходимо выполнить. На множестве операций определяется отношение  $Pos$  - последовательности выполнения операций представляемое, в виде ориентированного бесконтурного графа  $(Op, E)$  с множеством  $Op$  вершин, соответствующих операциям, и множеством  $E \subseteq Op \times Op$  дуг. При этом  $Op_i Pos Op_j \Leftrightarrow (Op_i, Op_j) \in E$ .

Для  $i$ -ой операции задается длительность выполнения  $\sigma_{ij} > 0$  на  $j$ -м типе ресурса.

Каждая операция характеризуется потребностью  $r_{ij}$  в ресурсах типа  $j$ , причем  $r_{ij} \leq r_0^j$ .

Удельная стоимость (или вес)  $w_i$  операции чаще всего полагается постоянной. Тогда

стоимость завершения операции равна  $w_i f_i$  и представляет собой

неубывающую от срока завершения операции функцию. Параметры  $\sigma_{ij}$ ,  $r_{ij}$  и  $w_i$  образуют вектор параметров операции, а совокупность таких векторов параметрически описывает систему операций и соответствующую графовую модель.

Еще одним критерием являются ограничения на составление расписаний. Расписание (1) является допустимым, если выполняются следующие условия:

$$(\forall Op_i \in Op) f_i \leq T, \quad (3)$$

$$(\forall Op_i, Op_j \in Op) Op_i Pos Op_j \Rightarrow f_i \leq s_j, \quad (4)$$

$$(\forall t \in [0, T], Op_t = \{Op_i \in Op \mid s_i \leq t \leq f_i\})$$

$$r^j(t) = \sum_{Op_i \in Op_t} r_{ij} \leq r_t^j, \quad (5)$$

где  $Op$  - множество операций, выполняемых в момент времени  $t$ .  $r^j(t)$  - потребность в ресурсе  $j$ -го типа, а  $r^i$  - уровень наличия ресурса  $j$ -го типа в момент времени  $t$ . Исходя из вышеописанных принципов формулируется задача составления допустимого в соответствии с (3)-(5) расписания (1).

Решим поставленную задачу для нашего кластера нейрокомпьютеров. Отметим, что в данной статье не рассматривается алгоритм выбора вычислительной структуры нейрокластера под задачу, данный вопрос рассматривается в работе [7].

Вычислительная структура, полученная в результате работы алгоритма, описанного в [7], по своей структуре является полным взвешенным направленным графом. В этом случае ее удобно представить в виде матрицы смежности, строки и столбцы которой будем именовать

коэффициентами  $i, j \in N$ .

### Алгоритм динамического формирования вычислительной структуры

Алгоритм динамического формирования вычислительной структуры:

1. Считывание очередной подзадачи из очереди задач. Если таких не осталось, то переходим на шаг 4.

2. Считывание строки матрицы смежности вычислительной структуры.

· Если подпрограммы, соответствующие операциям в структуре находятся в отношении параллельности, то возвращаемся на шаг 1.

· Если подпрограммы, находятся в отношении последовательности, то переходим на шаг 3.

3. Направление рекурсивного запроса от диспетчера кластера  $\Delta$  в сетевую топологию, созданную программным путем, т.е. совершаем обход матрицы смежности топологии вычислительной сети кластера. Хвост пакета запроса содержит совокупность параметров

$\{C_i, M_i, D_i, N\}, i = \overline{1, N}$ , где  $C_i, i = \overline{1, N}$ ,  $M_i, i = \overline{1, N}$ ,  $D_i, i = \overline{1, N}$ , определяющие количество нейропроцессоров, объемы свободной оперативной памяти и дисковой памяти в

мегабайтах,  $N_i, i = \overline{1, N}$  - разрядность данных, которые возможно разместить на каждом нейропроцессоре вычислительного узла.

4. Резервирование вычислительной машины под каждую подзадачу соответственно очереди, затем направление подзадачи и данных для их работы на зарезервированных вычислительные машины, затем вычисление результатов операций, затем сигнализирование об окончании вычислений пакетом, содержащим результаты вычислений в диспетчер кластера  $\Delta$ , переход на шаг 1, затем конец вычислений, выдача финального результата вычислений.

Таким образом, был совершен переход от топологии вычислительного кластера к вычислительной нейрокластерной структуре.

Рассмотрен способ программного виртуального эмулирования кластерной топологии, на примере топологии четырехмерного куба, более высокого уровня, а затем и динамического резервирования в полученной топологии машин для создания вычислительной нейрокластерной структуры. Данный процесс позволяет реализовать основную идею разработки кластера на нейрокомпьютерах. В основе его лежит принцип двойного распараллеливания задач, поступающих на выполнение в такой кластер. Первый этап - это стандартное разбиение задач на атомарные подзадачи, второй этап - размещение на

нейрокомпьютеры кластера такого количества задач и данных для них, чтобы их выполнение не занимало более одного нейропроцессорного такта.

### **Список литературы:**

1. Бурцев В. С. Параллелизм вычислительных процессов и развитие архитектуры супер ЭВМ. М.: ИВВС РАН, 1997. 152 с.
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
3. Галушкин А.И. Нейронные сети: основы теории. М.: «Горячая линия Телеком», 2010. 496 с
4. Лукашенко В.В. Анализ основных вопросов классификаций распределенных вычислительных систем // Современная техника и технологии. 2015. № 4 (44). С. 65-69.
5. Лукашенко В.В. Математическая модель реструктуризуемого под классы задач, виртуализуемого кластера вычислительной grid-системы на базе нейропроцессоров // В сборнике: Наука, Техника, Инновации 2014 сборник статей Международной научно-технической конференции. Под общей редакцией А.Л. Сафонова. 2014. С. 232-236.
6. Ручкин В.Н., Романчук В.А., Лукашенко В.В. Обобщенная модель вычислений кластера нейрокомпьютеров // Вестник Рязанского государственного университета им. С.А. Есенина. 2015. № 2 (47). С. 146-150.
7. 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.
8. Ручкин В.Н., Романчук В.А., Лукашенко В.В. Модель вычислений реструктуризуемого под задачу кластера на базе нейрокомпьютеров // Современная техника и технологии. 2015. № 3 (43). С. 56-59.
9. Топорков В.В. Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.