

**ЛУКАШЕНКО В. В., РОМАНЧУК В. А., РУЧКИН В. Н.
РАЗРАБОТКА АЛГОРИТМА ЭМУЛИРОВАНИЯ И ОПТИМИЗАЦИИ
ТОПОЛОГИИ И ДИНАМИЧЕСКОГО РЕЗЕРВИРОВАНИЯ РЕСУРСОВ
ВЫЧИСЛИТЕЛЬНОЙ СЕТИ КЛАСТЕРА НЕЙРОКОМПЬЮТЕРОВ**

УДК 004.75 ВАК 05.13.18, ГРНТИ 50.39.15.

Разработка алгоритма эмулирования
и оптимизации топологии
и динамического резервирования
ресурсов вычислительной сети кластера
нейрокомпьютеров

Development of the emulation
algorithm and the topology
optimization and dynamic resource
reservation network, the cluster of
Neurocomputers

В. В. Лукашенко, В. А. Романчук,
В. Н. Ручкин

V. V. Lukashenko,
V. A. Romanchuk, V. N. Ruchkin

Рязанский государственный
университет имени С. А. Есенина,
г. Рязань

Ryazan State University named
for S. Yesenin, Ryazan'

В статье рассмотрен способ программного виртуального эмулирования и оптимизации кластерной топологии более высокого уровня, а затем, и динамического резервирования в полученной топологии машин для создания вычислительной нейрокластерной структуры. Работа выполнена при финансировании государственного задания, проект №2.9519.2017/8.9

The article discusses the algorithm for software virtual emulation and optimization of a cluster topology of a higher level, and then, and dynamic reservation in the resulting topology of machines for creating a computational neurocluster structure. The work was carried out with the financing of the state task, project No. 2.959.2017/8.9

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

Keywords: distributed computing, cluster computing, clustering of computing resources, neurocomputing, 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 «НТИЦ Модуль», технические характеристики, преимущества и перспективность которых описана в работах [2–8].

Отличительной чертой такого вычислительного кластера является нестандартная архитектура вычислительных узлов. Нейрокомпьютеры – это ЭВМ нового, 6-го поколения, отличающиеся от классических вычислительных систем параллельного типа тем, что для решения задач они используют результаты функционирования множества искусственных нейронных сетей (ИНС), обученных с использованием специально подобранных исходных данных [3].

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. Следует отметить, что в данной статье топологии кластерных систем рассматриваются только с позиции программной реализации, т. е. инструментально вычислительная сеть кластера может быть организована по стандартной Ethernet технологии. 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.

Теоретический анализ

Рассмотрим типовые топологии кластерных вычислительных сетей. В литературе выделяется достаточно большое число топологий, таких как плоская двумерная решетка, трехмерный куб, четырехмерный куб, полносвязное кольцо, толстое дерево и другие [9]. Основной принцип использования и оптимизации нестандартных топологий вычислительных сетей состоит в том, что если в системе максимальное расстояние между любыми крайними компьютерами больше 4, то такая система не может работать эффективно [9]. Для получения более компактной конфигурации необходимо решить задачу о нахождении фигуры, имеющей максимальный объем при минимальной площади поверхности. В трехмерном пространстве таким свойством обладает шар. Но поскольку необходимо построить узловую систему, вместо шара приходится использовать куб (если число процессоров равно 8) или гиперкуб, если число процессоров больше 8. Размерность гиперкуба будет определяться в зависимости от числа процессоров, которые необходимо соединить. Архитектура гиперкуба является не самой эффективной, но самой наглядной, ее и будем использовать в данной работе.

Экспериментальное исследование

Рассмотрим с алгоритмической точки зрения процесс организации и оптимизации четырехмерного гиперкуба на базе стандартной Ethernet сети.

1. Выбор первой вычислительной машины кластерной подсети, у которой четвертая секция IP-адреса равна «1».

2. Направление broadcast-запроса на широковещательный адрес вычислительной подсети.

3. Рекурсивный просмотр tcpdump-файлов каждой вычислительной машины, поиск четырех наименьших значений времени приема пакета.

4. Запись в матрицу смежности в соответствующую ячейку последней четвертой секции IP-адреса.

5. Переход на шаг 1, выбирая вычислительную машину со следующим порядковым номером 4 секции IP-адреса.

Пример. Пусть есть вычислительный кластер K с количеством вычислительных машин $p_i, i = \overline{1,10}$. 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++ (рис. 1).

```

6 void SendBroadcast(char * message) {
7     SOCKET s = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
8     if(s == -1) {
9         printf("Error in creating socket");
10        return;
11    }
12    char opt = 1;
13    setsockopt(s, SOL_SOCKET, SO_BROADCAST, (char*)&opt, sizeof(char));
14    SOCKADDR_IN brdcastaddr;
15    memset(&brdcastaddr, 0, sizeof(brdcastaddr));
16    brdcastaddr.sin_family = AF_INET;
17    brdcastaddr.sin_port = htons(MY_PORT);
18    brdcastaddr.sin_addr.s_addr = INADDR_BROADCAST;
19    int len = sizeof(brdcastaddr);
20
21    char sbuf[1024];
22    //Заполнение sbuf здесь
23
24    sendto(s, sbuf, (int)strlen(sbuf), 0, (sockaddr*)&brdcastaddr, len);
25    closesocket(s);
26 }

```

Рисунок 1. Функция C++, реализующая отправку пакета на широковещательный адрес

Утилита tcpdump на каждой из машин кластера выдаст следующую информацию (рис. 2). Где самым важным параметром для нас является время, в данном случае 12:16:29.465780.

```

sh-4.2$ tcpdump -ne -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:16:29.465780 0a:00:27:00:00:00 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806),
length 42: Request who-has 192.168.0.1 tell 192.168.0.9, length 28

```

Рисунок 2. Работа утилиты tcpdump

Формируем очередную ячейку матрицы смежности связей вычислительной подсети кластера. В финальном варианте работы алгоритма матрица примет следующий вид (табл. 1).

Таким образом, был показан программный переход от технологии Ethernet к топологии четырехмерного гиперкуба (рис. 3).

Таблица 1. Матрица смежности связей топологии четырехмерный гиперкуб для вычислительной сети из 10 машин

K	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	1	2	3	0	5	0	0	0	0	0
p_2	1	2	0	4	0	6	0	0	0	0
p_3	1	0	3	4	0	0	0	8	0	10
p_4	0	2	3	4	0	0	7	0	9	0
p_5	1	0	0	0	5	6	0	8	0	0
p_6	0	2	0	0	5	6	7	0	0	0
p_7	0	0	0	4	0	6	7	8	0	0
p_8	0	0	3	0	5	0	7	8	0	0
p_9	0	0	0	4	0	0	0	0	9	10
p_{10}	0	0	3	0	0	0	0	0	9	10

Следующим шагом исследования является динамическое построение вычислительной нейрокластерной структуры из топологии четырехмерного куба. Данный процесс в вычислительном кластере обеспечивает планировщик ресурсов. Планировщиком ресурсов называется алгоритм, выходным результатом ко-

того является расписание резервирования и направления на вычисления очередной задачи. Рассмотрим модель расписания и критерии по которым составляется расписание планировщиком ресурсов.

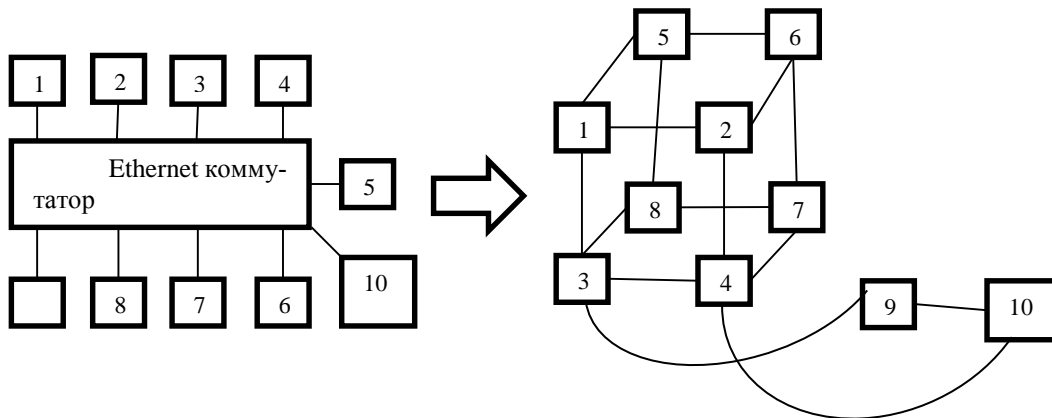


Рисунок 3. Программный переход от Ethernet технологии к топологии четырехмерного куба

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 $Op_i, i = \overline{1, N}$. 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 $Op = \langle Op_1, Op_2, \dots, Op_p, \dots, Op_p \rangle$ 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 , при-

чем $r_{ij} \leq r_0^i$. Удельная стоимость (или вес) w_i операции чаще всего полагается постоянной. Тогда стоимость завершения операции равна $w_i f_i$ и представляет собой функцию, представляющую собой неубывающую от срока завершения операции функцию. Параметры $\sigma_{i,j}$, 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:

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

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

$$\begin{aligned} (\forall t \in [0, T], Op_t = \{Op_i \in Op \mid s_i \leq t \leq f_i\}) \\ r^j(t) = \sum_{Op_i \in Op} r_{ij} \leq r_t^i, \end{aligned} \quad (5)$$

где Op_t – множество операций, выполняемых в момент времени t ; $r^j(t)$ – потребность в ресурсе j -го типа, а r_t^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 8.
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 Δ 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.

5. Направление подзадачи и данных для их работы на зарезервированных вычислительные машины.

6. Вычисление результатов операций.

7. Сигнализирование об окончании вычислений пакетом, содержащим результаты вычислений в диспетчер кластера Δ . Переход на шаг 1.

8. Конец вычислений, выдача финального результата вычислений.

Таким образом, был совершен переход от топологии вычислительного кластера к вычислительной нейрокластерной структуре (рис. 4).

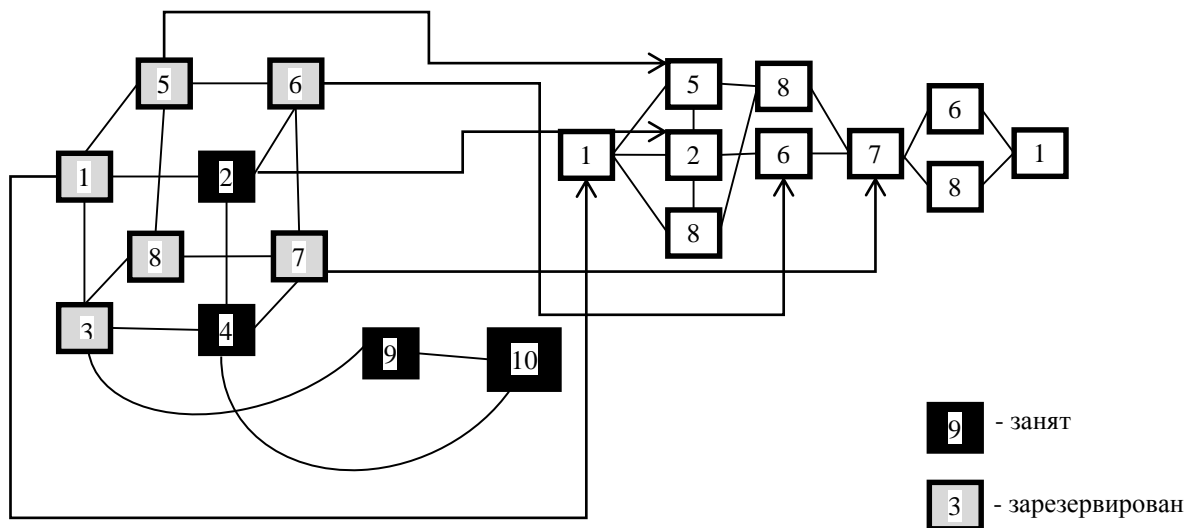


Рисунок 4. Схема процесса резервирования вычислительных машин в топологии гиперкуба для вычислительной структуры нейрокластера

Вывод

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 / Под общей редакцией А. Л. Сафонова // Наука, Техника, Инновации 2014 : сборник статей Международной научно-технической конференции. 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.

List of references

1. Burcev, V. S., *Parallelism of computational processes and the development of architecture in super computers*, Moscow: IAF RAS, 1997, 152 p.
2. Voevodin, V. V., Voevodin, Vl. V., *Parallel computing*, Saint Petersburg: BHV-Peterburg, 2002, 608 p.
3. Galushkin, A. I., *Neural network: basic theory*, Moscow: Hotline Telekom, 2010, 496 p.
4. Lukashenko, V. V., “Analysis of the main classifications of distributed computing systems”, *Modern equipment and technology*, 2015, no. 4 (44), pp. 65–69.
5. Lukashenko, V. V., “A mathematical model restructurizing under classes of problems, virtualize with heater cluster computing grid-system on the basis of neuroprocessor”, *Science, Technology, Innovation 2014*, collection of articles of International scientific-technical conference. Under the General editorship A. L. Safonova, 2014, pp. 232–236.

6. Ruchkin, V. N., Romanchuk, V. A., Lukashenko, V. V., “A generalized computing model of a cluster of Neurocomputers”, *Bulletin of the Ryazan state University. S. A. Esenin*, 2015, no. 2 (47), pp. 146–150.

7. Ruchkin, V. N., Romanchuk, V. A., Fulin, V. A., Lukashenko, V. V., “The algorithm of choice of computational structure distributed cluster neurocomputer architecture”, *Izvestiya of the Tula state University. Engineering science*, 2015, no. 9, pp. 236–244.

8. Ruchkin, V. N., Romanchuk, V. A., Lukashenko, V. V., “The computing model restructurizing under task cluster on the basis of neuro-computers”, *Modern equipment and technology*, 2015, no. 3 (43), pp. 56–59.

9. Toporkov, V. V., *Models of distributed computing*, Moscow: FIZMATLIT, 2004, 320 p.