

ПАРАЛЛЕЛЬНАЯ ГРУППОВАЯ ПЕРЕДАЧА «ВСЕ — ВСЕМ» В РАСПРЕДЕЛЕННОМ ПОЛНОМ КОММУТАТОРЕ

В.С. Подлазов

Рассмотрен способ параллельного выполнения групповой передачи «все — всем» в распределенных полных коммутаторах, построенных на базе комбинаторных блок-схем, для многопроцессорных вычислительных систем.

Ключевые слова: многопроцессорная вычислительная система, произвольная перестановка пакетов данных, параллельная бесконфликтная самомаршрутизация, симметричная блок-схема, распределенный полный коммутатор, групповая передача «все — всем».

ВВЕДЕНИЕ. ПОСТАНОВКА ЗАДАЧИ

За сетями связи многопроцессорных вычислительных систем (МВС) утвердился термин системные сети. Наименьшие времена передачи данных имеет та системная сеть (СС), которая обеспечивает прямые каналные соединения (без промежуточной буферизации данных) для любой пары абонентов сети при параллельной передаче от всех абонентов. Если такая СС позволяет бесконфликтно осуществлять произвольную перестановку пакетов данных по прямым каналам между абонентами посредством их независимой самомаршрутизации каждым абонентом, то она является неблокируемой и самомаршрутизируемой.

Этим свойством обладает, например, СС со структурой полного графа или состоит из одной СБИС полного коммутатора. Однако обе эти структуры непригодны для создания СС с большим числом абонентов — первая вследствие большого числа портов абонентов и каналов между ними, а вторая из-за невозможности создания коммутаторных СБИС с необходимым числом портов.

В работах [1–3] предложен метод расширения полного коммутатора $m \times m$ с сохранением его маршрутных свойств — метод построения распределенного полного коммутатора $N \times N$, построенного из малых коммутаторов $m \times m$, $m \ll N$. Этот коммутатор, обозначаемый как простейшая системная сеть $ПСС(N, m, 1)$, обладает свойствами неблокируемости и самомаршрутизируемости, но использует много меньше каналов (примерно

в $\sqrt{N}/2$ раз), чем полный коммутатор со структурой полного графа.

Этот метод основывается на описании СС в терминах симметричных блок-схем, давно уже исследуемых в комбинаторике [4]. Его можно применять каскадно, что позволяет строить многокаскадные распределенные полные коммутаторы с любым числом портов [3], которые являются неблокируемыми и самомаршрутизируемыми.

Распределенный полный коммутатор представляет собой практически идеальную СС для построения отказоустойчивых МВС реального времени [5], так как обеспечивает процессорную k -отказоустойчивость по процессорам при любом $k < N/2$, где N — число процессоров.

Групповые передачи «один — многим», «многие — одному» и «многие — многим» широко востребованы в МВС [6, 7]. Для построения отказоустойчивых МВС особенно востребована групповая передача «все — всем», обеспечивающая работоспособность МВС в условиях враждебных отказов многих процессоров [8].

В работах [9, 10] рассмотрен способ выполнения групповой передачи на СС вида $ПСС(N, m, 1)$ за время последовательной передачи $\sim m^2$ пакетов. Однако он неприменим для системных сетей, в которых $N \gg m(m-1) + 1$ и, в частности, для многокаскадного полного коммутатора.

В настоящей работе ставится задача разработки способа быстрой реализации групповой передачи «все — всем» в многокаскадном распределенном полном коммутаторе. Для $ПСС(N, m, 1)$ он должен быть быстрее способа из работ [9, 10].

1. РАСПРЕДЕЛЕННЫЙ ПОЛНЫЙ КОММУТАТОР

Неполной уравновешенной блок-схемой $B(N, M, m, k, \sigma)$ называется набор из M блоков, в которые входят N элементов так, что в каждый блок входит k разных элементов, каждый элемент входит в m блоков, а каждая пара элементов входит в σ блоков.

В соответствии с этим определением параметры блок-схемы связаны соотношениями: $Nm = kM$ и $N = m(k - 1)/\sigma + 1$. Минимальное число блоков M и максимальное число элементов N обеспечивается в случае симметричных блок-схем $B(N, m, \sigma)$, в которых $k = m$ и $M = N$. В дальнейшем рассматриваются симметричные блок-схемы $B(N, m, 1)$.

Теперь в определении блок-схемы $B(N, m, \sigma)$, заменим блок на некоторый исходный полный коммутатор $m \times m$ с m дуплексными портами — ИК(m), элемент — на процессор с m портами, входение элемента в блок — на дуплексное подключение абонента к одному из портов ИК(m) и, наконец, симметричную блок-схему — на СС с N абонентами — СС(N). Тогда последняя состоит из N копий ИК(m), к каждой копии подсоединено m разных процессоров, каждый процессор подсоединен к m копиям ИК(m), а между каждой парой процессоров имеется σ прямых каналов, которые проходят через разные ИК(m). При этом каждый канал *последовательно* проходит только через один ИК(m). Поэтому СС(N) наследует маршрутные свойства ИК(m). Такая СС называется простейшей — ПСС(N, m, σ). Очевидно, что ПСС(N, m, σ) изоморфна блок-схеме $B(N, m, \sigma)$. В этой работе рассматриваются только блок-схемы $B(N, m, 1)$ и ПСС($N, m, 1$).

Структура ПСС($N, m, 1$) имеет вид симметричного двудольного графа, в котором число узлов в каждой доле равно N , а степень любого узла в каждой доле равна m (рис. 1). Узлами одной доли являются ИК(m), а узлами другой доли — m -портовые процессоры A_i . Между любыми двумя узлами в каждой доле существует ровно 1 путь длины 2, про-

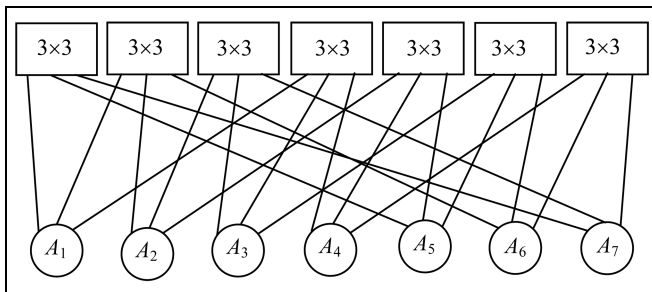


Рис. 1. Структура ПСС(7, 3, 1) в виде минимального квазиполного графа

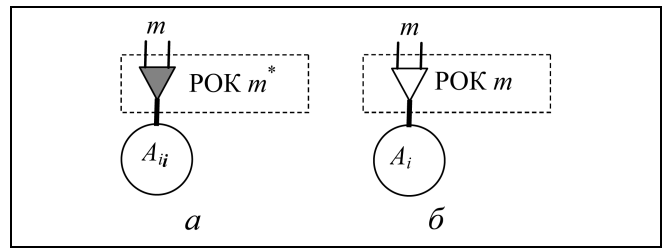


Рис. 2. Разветвитель/объединитель m дуплексных каналов РОК m

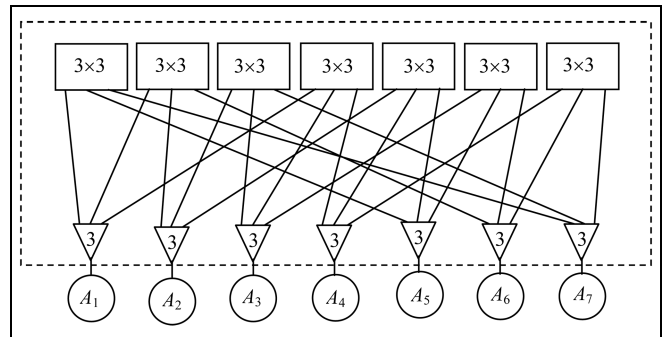


Рис. 3. Распределенный полный коммутатор 7×7 — РК $_1(7, 3, 1)$

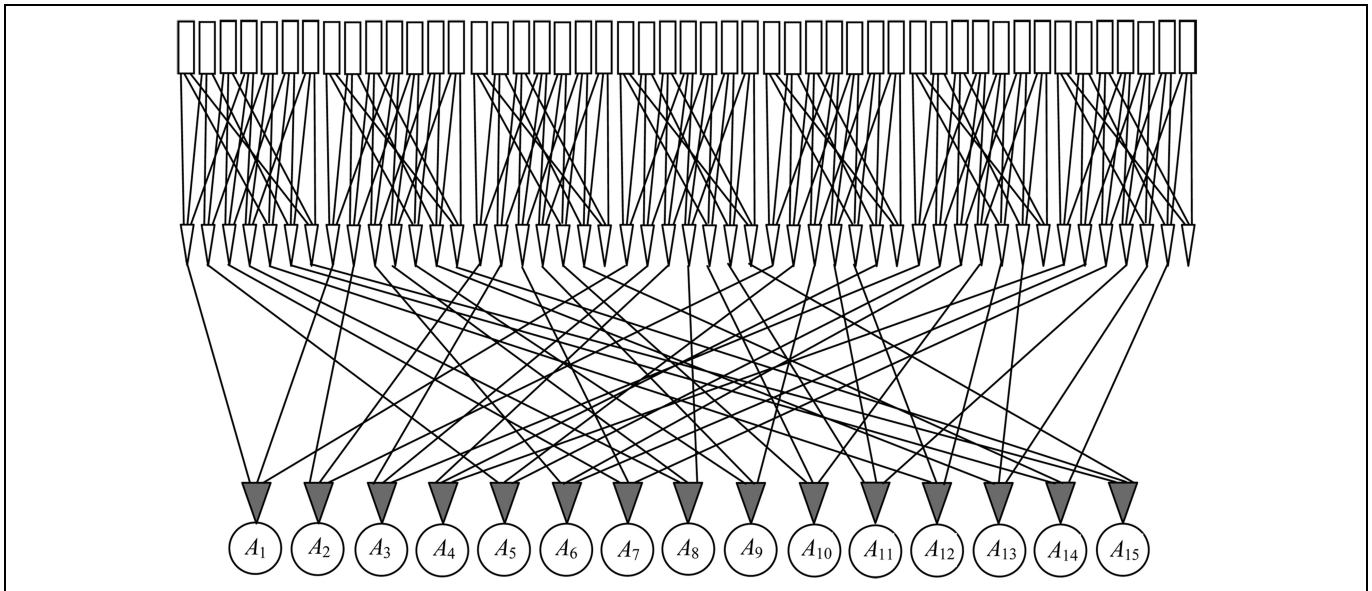
ходящий через узел в другой доле. Такой граф мы называем минимальным квазиполным графом [8].

В ПСС($N, m, 1$) можно заменить m -портовый процессор (рис. 2) на связку 1-портового процессора и разветвителя/объединителя каналов $(1 \times m)/(m \times 1)$ — РОК m . Будем различать два вида таких схем. Первый из них (рис. 2, а) позволяет всем m портам работать параллельно и независимо как на передачу, так и на прием. Схема обозначается РОК m^* и изображается с заливкой. Это схема типа многопортовой сетевой карты в PCI-Express [11, 12].

Второй вид-схема РОК m (рис. 2, б) — позволяет передавать только один пакет к процессору. Для этого она должна выполнять функцию множественного доступа — пропускать из нескольких пакетов только любой один, а остальные сбрасывать. Фактически это схема демультиплексора-мультиплексора, дополненная схемой множественного доступа в мультиплексоре.

Теперь схему ПСС($N, m, 1$) можно преобразовать в схему с двумя коммутирующими каскадами — хребта коммутаторов $m \times m$ и каскада РОК m . Пример такой схемы при $m = 3$ приведен на рис. 3.

Часть такой схемы (см. рис. 3), располагающаяся выше интерфейса «процессор — РОК m » и заключенная в пунктирный прямоугольник, представляет собой полный распределенный коммутатор $N \times N$ с N дуплексными портами, который является однокаскадным (по числу каскадов РОК m) распределенным коммутатором РК $_1(N, m, 1)$.


Рис. 4. Схема двухкаскадного полного коммутатора $PK_2(15, 3, 1|3)$

Если взять распределенный коммутатор $PK_1(N, m, 1)$ в качестве ИК(N), то, применяя ПСС($N, m, 1$) и РОК m , можно построить [3] двухкаскадный распределенный полный коммутатор $R_2 \times R_2$, в схемном базисе коммутаторов $m \times m$ и РОК m . Он обозначается как $PK_2(R_2, m, 1|m)$, и в нем число

дуплексных портов R_2 находится в диапазоне $N[N/m] < R_2 < N[N/m]$. В $PK_2(R_2, m, 1|m)$ процессоры, номера которых не совпадают по $\text{mod } m$, связаны одним путем, а которые совпадают по $\text{mod } m$, — m путями через разные $PK_1(N, m, 1)$.

Итеративное повторение этой процедуры позволяет построить k -каскадный полный коммутатор $PK_k(R_k, m, 1|m^{k-1})$ с R_k дуплексными портами, где $R_{k-1}[N/m] < R_k < R_{k-1}[N/m]$ [1, 3]. Он сохраняет маршрутные свойства полного коммутатора, т. е. представляют собой неблокируемую и самомаршрутизируемую СС и состоит из коммутаторов $m \times m$ и РОК m .

Не вдаваясь в подробности, проиллюстрируем построение $PK_2(15, 3, 1|3)$, применяя ПРС(7, 3, 1) и $PK_1(7, 3, 1)$. Табл. 1 задает схему межсоединений для них, которая соответствует рис. 2 и 3. Табл. 2 задает схему межсоединений для $PK_2(15, 3, 1|3)$, в которой строки задают разные $PK_1(7, 3, 1)$, а трехстолбцовые области — ПРС(7, 3, 1). В свою очередь, рис. 3 задает коммутатор $PK_2(15, 3, 1|3)$ в схемном базисе 3×3 и РОК3.

Таблица 1

Таблица межсоединений в $PK_1(7, 3, 1)$

ПРС(7, 3, 1) или $PK_1(7, 3, 1)$			
Номера ИК(3)	Номера подсоединенных к ИК(3) абонентов		
1	1	7	5
2	2	1	6
3	3	2	7
4	4	3	1
5	5	4	2
6	6	5	3
7	7	6	4

Таблица 2

Таблица межсоединений в $PK_2(15, 3, 1|3)$

$PK_1(7, 3, 1)$	1-я ПСС(7, 3, 1)			2-я ПСС(7, 3, 1)			3-я ПСС
1	1	7	5	8	14	12	15
2	2	1	6	9	8	13	15
3	3	2	7	10	9	14	—
4	4	3	1	11	10	8	15
5	5	4	2	12	11	9	—
6	6	5	3	13	12	10	—
7	7	6	4	14	13	11	—

2. ГРУППОВАЯ ПЕРЕДАЧА «ВСЕ — ВСЕМ»

Групповую передачу в $PK_k(R_k, m, 1|m^{k-1})$ очень просто можно реализовать, если каждый процессор передаст свой пакет в последовательности $(R_k - 1)$ -й смежных перестановок-транспозиций, в которой $(i + 1)$ -я перестановка получается циклическим сдвигом приемников на одну позицию от-



носителю их размещения в i -й перестановке. Такую реализацию можно ускорить в m раз, если использовать в первом каскаде схему РОК m^* (см. рис. 2, а), которая позволяет передавать и принимать разные пакеты независимо и параллельно.

В простейшем случае при $k = 1$ групповая передача реализуется в РК $_1(N, m, 1)$ следующим образом.

В i -м сеансе, $1 \leq i \leq m$, каждый процессор передает свой пакет через РОК m^* параллельно на все подсоединенные коммутаторы $m \times m$. В i -м сеансе каждый коммутатор принятый на j -й порт пакет передает на $[(i + j) \bmod m + 1]$ -й выходной порт. Если в любом сеансе принятый пакет совпадает с переданным, то процессор его сбрасывает. Длительность i -го сеанса равна времени передачи одного пакета.

Лемма 1. В результате i -го сеанса каждый процессор получит $m - 1$ разных пакетов от других процессоров.

Доказательство. Допустим противное, что на некоторый процессор-приемник поступают два одинаковых пакета от одного процессора-источника. Они поступают с разных коммутаторов, так как РОК m^* в любом сеансе разводит одинаковые пакеты по разным коммутаторам. Это означает, что между процессором-источником и процессором-приемником имеются два пути через разные коммутаторы, что в РК $_1(N, m, 1)$ невозможно по построению. Поэтому любой процессор получает не более $m - 1$ разных пакетов, так как он соединен только с $m - 1$ другими процессорами, а свой пакет он при приеме сбрасывает. Наличие конфликтов на выходных портах коммутаторов может привести к уменьшению числа пакетов, поступающих к процессорам. Но конфликтов быть не может, так как номера выходных портов повторяют номера входных портов с приращением на i единиц. Лемма доказана.

Теорема 1. За m сеансов каждый процессор получит $m(m - 1) = N - 1$ разных пакетов от других процессоров, т. е. выполнится групповая передача «все — всем» за время последовательной передачи m пакетов.

Доказательство. В каждом сеансе каждый процессор передает свой пакет ко всем коммутаторам, с которыми он соединен. Каждый коммутатор во всех m сеансах передает каждый поступивший пакет по всем каналам, кроме каналов поступления. В результате каждый процессор передает свой пакет ко всем другим процессорам, с которыми он соединен через коммутаторы, а в РК $_1(N, m, 1)$ он соединен таким образом одним каналом с любым другим процессором. Согласно лемме 1 во всех m сеансах каждый процессор получит $m(m - 1)$ пакетов и все они разные. Теорема 1 доказана. ♦

Аналогичным образом можно выполнять групповую передачу и в k -каскадном полном распределенном коммутаторе РК $_k(R_k, m, 1|m^{k-1})$. Для этого первый каскад РОК m , смежный с процессорами, должен содержать схемы РОК m^* , которые позволяют передавать и принимать разные пакеты независимо и параллельно (см. рис. 2, а), а все ос-

тальные каскады — схемы РОК m , которые позволяют передавать и принимать только один пакет (см. рис. 2, б и 4).

Групповую операцию «все — всем» на коммутаторе РК $_k(R_k, m, 1|m^{k-1})$ можно проводить как последовательность R_k разных перестановок, являющихся циклическими транспозициями. Перестановки из этой последовательности можно запускать параллельно по m штук на разных коммутаторах РК $_{k-1}(R_{k-1}, m, 1|m^{k-2})$, как параллельно выполняемые последовательности R_{k-1} разных перестановок-транспозиций среди процессоров, подсоединенных к каждому коммутатору РК $_{k-1}(R_{k-1}, m, 1|m^{k-2})$.

Простая реализация этой идеи состоит в следующем. Номер сеанса i представляется k -разрядным числом в m -ичной системе счисления как $i = m^{k-1}a_k + m^{k-2}a_{k-2} + \dots + a_1 + 1$, и групповая передача реализуется следующим образом.

В i -м, $1 \leq i \leq m^k$, сеансе каждый процессор передает свой пакет через схемы РОК m^* первого каскада параллельно на все подсоединенные коммутаторы РК $_{k-1}(R_{k-1}, m, 1|m^{k-2})$. Каждая схема РОК m r -го, $1 < r < k$, каскада передает пакет на $(a_r + 1)$ -й выходной порт, считая что порты нумеруются натуральными числами. Каждый коммутатор пакет, принятый на j -й порт, передает на $[(a_k + 1 + j) \bmod m + 1]$ -й порт. Если в любом сеансе принятый пакет совпадает с переданным, то процессор его сбрасывает. Длительность i -го сеанса равна времени передачи одного пакета.

Лемма 2. В результате i -го сеанса каждый процессор получит не более $m - 1$ разных пакетов от других процессоров.

Доказательство. Наличие в РК $_k(R_k, m, 1|m^{k-1})$ нескольких путей между любой парой процессоров может привести к получению в некотором сеансе одинаковых пакетов. Они поступают с разных коммутаторов в РК $_{k-1}(R_{k-1}, m, 1|m^{k-2})$ и схема РОК m^* в любом сеансе их пропускает к процессору. Аналогичные ситуации, возникающие в коммутаторах РК $_{k-s}(R_{k-s}, m, 1|m^{k-s-1})$, $1 < s < k$, разрешаются схемами множественного доступа в РОК m . Любой процессор принимает пакеты только от других процессоров, так как свои пакеты он сбрасывает.

Теорема 2. Каждый процессор за m^k сеансов получит $R_k - 1$ разных пакетов от других процессоров, т. е. выполнится групповая передача «все — всем» за время последовательной передачи m^k пакетов.

Доказательство. В каждом сеансе каждый процессор передает свой пакет ко всем коммутаторам РК $_{k-1}(R_{k-1}, m, 1|m^{k-2})$, с которыми он соединен. Каждый такой коммутатор во всех сеансах передает каждый поступивший пакет по всем своим каналам. В результате

каждый процессор передает свой пакет ко всем другим процессорам, с которыми он соединен через коммутаторы $R_{k-1}(R_{k-1}, m, 1|m^{k-2})$, а в $R_k(R_k, m, 1|m^{k-1})$ он соединен таким образом с любым другим процессором. В результате реализуется групповая передача «все — всем» с возможными приемами одинаковых пакетов. Это пакеты от других процессоров, так как свои пакеты любой процессор сбрасывает. ♦

Замечание. При $k > 1$ для данного способа справедливо соотношение $m^k > R_k/m$, т. е. на реализацию групповой передачи тратится больше времени, чем первоначально было задумано. Накладные затраты времени равны времени передачи $(m^k - R_k/m)$ пакетов. Это составляет $m^k - R_k/m \approx m^k - (m-1) \times (m-1 + 1/m)^k/m < m^k - (m-1)m^{k-1} = m^{k-1}$, т. е. не более $(1/m)$ -й времени выполнения групповой операции.

3. ФОРМИРОВАНИЕ МАРШРУТНЫХ АДРЕСОВ

Самомаршрутизация в $R_k(R_k, m, 1|m^{k-1})$ предполагает, что каждый источник независимо от других абонентов формирует цуг маршрутных адресов, прописывающий бесконфликтный путь через все каскады к приемнику. Цуг имеет вид $(C_1, C_2, \dots, C_k, B)$, где $C_r, 0 \leq C_r \leq m, 1 \leq r < k$, — номер выходного порта схемы POK_m r -го каскада, а $B (1 \leq B \leq m)$ — номер выходного порта коммутатора $m \times m$ в хребте.

Для выполнения групповой передачи «все — всем» в $R_1(N, m, 1)$ C_1 в любом сеансе задается как широкоэмитательный адрес, который может задаваться нулем: $C_1 = 0$. В i -м сеансе B задается как $B = (i + j) \bmod m + 1$.

Для выполнения групповой передачи «все — всем» в $R_k(R_k, m, 1|m^{k-1})$ достаточно иметь $C_1 = 0$, $C_r = (a_r + 1 + j) \bmod m + 1, 1 < r < k$, и $B = (a_k + 1 + j) \bmod m + 1$.

Напомним, что j в предложенной адресации обозначает номер порта, принимающего пакет в схемах POK_m r -го каскада или коммутатора $m \times m$. Это означает использование косвенной адресации относительно номера j . Рассматриваемый способ допускает прямую адресацию с явным заданием значения j . Но это требует задания схемы $R_k(R_k, m, 1|m^{k-1})$ с точным описанием номеров портов и составления на этой базе обширных таблиц маршрутизации. Фактически это означает возможность применения двух способов самомаршрутизации — динамической с косвенной адресацией и статической с прямой адресацией. Выбор — за разработчиками схем для $R_k(R_k, m, 1|m^{k-1})$.

ЗАКЛЮЧЕНИЕ

Предложен простой способ параллельного выполнения групповой передачи «все — всем» на многокаскадном распределенном полном коммутаторе.

Предполагаемое развитие данной работы — построение способа параллельного выполнения групповой передачи «все — всем» на многокаскадном распределенном коммутаторе с $\sigma > 1$ параллельными каналами между любой парой процессоров [2, 3]. Можно ожидать дальнейшего сокращения времени выполнения групповой передачи благодаря одновременному выполнению σ разных перестановок.

ЛИТЕРАТУРА

1. Подлазов В.С., Соколов В.В. Метод однородного расширения системных сетей многопроцессорных вычислительных систем // Проблемы управления. — 2007. — № 2. — С. 22–27.
2. Николаев А.Б., Подлазов В.С. Отказоустойчивое расширение системных сетей многопроцессорных вычислительных систем // Автоматика и телемеханика. — 2008. — № 1. — С. 162–170.
3. Каравай М.Ф., Подлазов В.С. Распределенный полный коммутатор как «идеальная» системная сеть для многопроцессорных вычислительных систем // Управление большими системами. — Вып. 34. — С. 92–116. — URL: <http://ubs.mtas.ru/upload/library/UBS3405.pdf> (дата обращения 29.08.2012).
4. Холл М. Комбинаторика. — М.: Мир, 1970. — Гл. 10–12.
5. Каравай М.Ф., Пархоменко П.П., Подлазов В.С. Универсальная сетевая структура для отказоустойчивых многопроцессорных систем реального времени // Тр. междунар. конф. «Технические и программные средства систем управления, контроля и измерения» (УКИ'10) / Москва, ИПУ РАН, окт. 2010. — М., 2010. — С. 79–91.
6. An Overview of the BlueGene/L Supercomputer. The BlueGene / L Team / N. Adiga, G. Almasi, G.S. Almasi, et al. — URL: http://www.linuxfordevices.com/files/article041/bgl_sc2002_final.pdf.
7. Стецюра Г.Г. Методы ускорения выполнения коллективных операций стандарта MPI // Проблемы управления. — 2005. — № 6. — С. 78–81.
8. Лобанов А.В. Обнаружение и идентификация неисправностей в распределенных управляющих вычислительных системах с программно-управляемой сброс- и отказоустойчивостью // Автоматика и телемеханика. — 1998. — № 1. — С. 155–164.
9. Youngjoo Cho, Changkyun Chi and Ilyong Chung. An Efficient Conference Key Distribution System Based on Symmetric Balanced Incomplete Block Design // Lecture Notes Comput. Sci. (LNCS) 2657. — 2003. — P. 147–154.
10. Okbin Lee, Sangho Lee, Seongyeol Kim and Ilyong Chung. An Efficient Load Balancing Algorithm Employing a Symmetric Balanced Incomplete Block Design // 3046. Ibid. — 2004. — P. 647–654.
11. Горбунов В.С. Архитектура хорошо масштабируемого вычислительного кластера // Тр. междунар. науч.-техн. конф. «Суперкомпьютерные технологии: разработка, программирование, применение» (СКТ-2010) / Дивноморское, сент. 2010. — Т. 1. — С. 48–54.
12. Горбунов В.С., Лацис А.О., Иванов А.Н. О построении суперкомпьютеров на основе интерфейса PCI-Express // Там же. — С. 55–57.

Статья представлена к публикации членом редколлегии чл.-корр. РАН П.П. Пархоменко.

Подлазов Виктор Сергеевич — д-р техн. наук, гл. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ☎ (495) 334-78-31, ✉ podlazov@ipu.ru.