

УДК 021.8 + 025.1
ББК 78.34

«РЕБЕРНОЕ» МАСШТАБИРОВАНИЕ ГИПЕРКУБИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ¹

Мелентьев В. А.²

*(ФГБУН Институт физики полупроводников
им. А.В. Ржанова СО РАН, Новосибирск)*

Исследуется проблема повышения потенциала распараллеливания задач в вычислительной системе без наращивания числа ее процессоров: модифицированием исходной топологии, дополняющим ее отношения смежности между процессорами, при том, что базовые способы адресации и маршрутизации остаются прежними. В работе обосновывается постановка задачи «реберного» масштабирования, состоящего в использовании избыточных в текущей конфигурации портов для создания линков между наиболее удаленными процессорами. Для систем с гиперкубической топологией получены функции предельной распараллеливаемости и топологической масштабируемости на задачах, в которых длина пути между информационно смежными ветвями лимитирована объемами обрабатываемых данных и используемой сетевой технологией. Приведены соответствующие графики, и дан их анализ.

Ключевые слова: гиперкубическая вычислительная система, суперкомпьютер, параллелизм, топологическая масштабируемость, реберное масштабирование.

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект №14-07-00169а.

² Виктор Александрович Мелентьев, кандидат технических наук (melva@isp.nsc.ru).

1. Введение

Известно, что ограничение ускорения и эффективности распараллеливания задачи при отсутствии в ней скалярных фрагментов обусловлено наименее быстродействующим компонентом реализующей эту задачу подсистемы [18], каковым, как правило, является коммуникационная среда – интерконнект вычислительной системы (ВС). Быстродействие этой среды обусловлено расстояниями между информационно смежными в задачах процессорами [5] и используемой сетевой технологией.

Центральной проблемой горизонтального масштабирования ВС, связанной с ожиданиями соразмерного приращению параллелизма решаемых задач, является то, что увеличение порядка информационного графа задачи приводит к неполиномиальному росту трудоемкости поиска изоморфных его вложений при существенном снижении вероятности наличия таковых. Это связано с тем, что в основе используемых при вложении параллельной задачи алгоритмов лежит сопоставление отношений смежности вершин информационного графа задачи и вершин графа системы. Однако в силу известных технико-экономических факторов полная связность всех процессоров системы реализована быть не может, поэтому условие физической смежности задействованных в параллельной задаче процессоров даже при малом их числе не всегда может быть реализовано, и с масштабированием задачи эта проблема лишь усугубляется.

Для решения этой проблемы в информационную модель ВС предложено представлять графами, отношения смежности вершин в которых удовлетворяют требованиям в части предельно допускаемых для параллельных задач расстояний δ между процессорами организованных для их решения подсистем [8]. Такие, порожденные множеством вершин V графа $G(V, E)$ и матрицей δ -достижимости \mathbf{R}_δ графы, названы графами δ -достижимости $G_\delta(V, E_\delta)$. В них $E_\delta \in \mathbf{R}_\delta$, где R_δ – отношение δ -достижимости, связывающее ребром вершины u и v , если расстояние $d(u - v)$ в графе $G(V, E)$ не превышает величины δ : $(u - v) \in R_\delta \Leftrightarrow d(u - v) \leq \delta$. В [10] установлены функциональные зависимости таких расстояний от свойственных задаче объемов вычислительных операций

W и обмениваемых данных Q , от требуемых значений ускорения S_p и/или эффективности E_p решения этих задач при их распараллеливании на p ветвей. Большому значению δ при этом соответствует большее значение степени $s(G_\delta)$ получаемого таким образом графа δ -достижимости G_δ и большее значение его δ -плотности¹ $\varphi(G_\delta) \equiv \varphi_\delta(G)$. Естественно, что вложение в такие графы параллельных задач, ранг которых не превышает δ -плотности, может быть произвольным, а возможности изоморфного вложения в них задач с большим числом ветвей – существенно выше. В [11] проведено исследование функциональной зависимости порядка наибольшей компоненты δ достижимости (δ -клики) гиперкубической ВС² от размерности s гиперкуба и от лимитируемой в нем достижимости δ вершин.

В [10] впервые восполнен пробел в оценке влияния топологии на масштабируемость параллельных систем и решаемых на них задач. С этой целью была предложена разделенная на две составляющие модель параллельных вычислений: 1-я часть отнесена к параллельным приложениям и приписывает им свойства неограниченной распараллеливаемости [1], 2-я отнесена к вычислительной системе, ограничения параллелизма в которой обусловлены дефицитным в сравнении с процессорами быстродействием интерконнекта. Это позволило разделить технологический и топологический аспекты масштабирования системы и исследовать взаимную их обусловленность при масштабировании задач в этой системе. При этом выявлено, что в зависимости от собственных задачи объемов вычислительных операций W и обмениваемых данных Q ускорение S_p ее решения, минимальное число p используемых при этом процессоров и лимитируемое при этом расстояние δ между информационно смеж-

¹ По аналогии с плотностью графа G его δ -плотность равна порядку наибольшей в графе δ -достижимости G_δ клики $K(G_\delta)$.

² Здесь и далее, говоря о ВС, по умолчанию предполагаем использование ею сетевой технологии, которая для (W, Q) -задачи ранга p с требуемыми значениями показателей эффективности ее решения обуславливает предельно допустимую достижимость δ .

ными процессорами коррелированы используемой в системе сетевой технологией (*Network Technology* – *NT*). Там же показано, что совместность определенных таким образом параметров для заданного набора (W, Q) -задач, решаемых в системе с заданной *NT*, определяется топологией ВС. Иными словами, при решении на ВС некоторой (W, Q) -задачи заданное ускорение S может быть обеспечено при разбиении этой задачи не менее чем на p параллельных ветвей, а так как расстояния между информационно смежными процессорами ограничены используемой в системе *NT* и не должны превышать некоторого значения $\delta(p)$, то топология должна обеспечить наличие в графе ВС соответствующего подграфа δ -достижимости порядка, не меньшего p .

Понятно при этом, что информационно полносвязные задачи предъявляют к топологии ВС наиболее жесткие требования, и система обладает наилучшей топологической масштабируемостью на топологически адекватных ей задачах. В [15] это наглядно продемонстрировано сопоставлением функций топологической масштабируемости гиперкубических ВС и предельной распараллеливаемости на них задач с полносвязной, кольцевой и звездной информационными топологиями.

Учитывая очевидную зависимость порядка полносвязной с лимитируемой достижимостью компоненты регулярного графа от его степени, в настоящей работе предложен способ «реберного» масштабирования, суть которого состоит в повышении потенциальных возможностей системы в распараллеливании задач, не изменяя число процессоров в ней, а существенно менее дорогостоящим повышением степени графа ВС, сохраняющим базовые (исходные) отношения смежности вершин. Необходимым условием при этом является требование использования характерных для исходной топологии ВС принципов адресации и алгоритмов маршрутизации.

В данной работе реберное масштабирование рассмотрено на примере систем с гиперкубической топологией. Сохранение присутствующих «гиперкубу» достоинств маршрутизации, основанной на тождественности расстояний между его вершинами и кодовых расстояний между их двоичными адресами, достигнуто введением дополнительной, инверсной адресации. Для модифицированного таким образом гиперкуба определены диаметр и ширина

бисекции, формализована функция δ -плотности; даны графические сопоставления этих характеристик и функций масштабированности для реберно масштабированного и исходного гиперкубов. Сделаны выводы о потенциале и эффективности предложенного способа масштабирования в сравнении с горизонтальным масштабированием гиперкубических ВС.

2. «Реберное» масштабирование вычислительной системы

2.1. ОБЩЕЕ ПОНЯТИЕ О РЕБЕРНОМ МАСШТАБИРОВАНИИ

Общеизвестные понятия «горизонтального» и «вертикального» масштабирования вычислительной системы различаются способами повышения интегральной ее производительности: наращиванием числа составляющих систему процессорных элементов или их заменой более производительными. В отличие от этих понятий, введем понятие «реберного» масштабирования вычислительной системы, состоящее в таком наращивании числа ребер в графе ВС, при котором остаются неизменными не только множество вершин и подмножество ребер исходного графа ВС, но и те базовые его свойства, на которых основана используемая в системе маршрутизация. Учитывая, что сетевые топологии ВС за исключением звездной, как правило, являются регулярными, далее предполагаем, что реберное масштабирование ВС состоит в таком изменении степени ее графа, при котором характерные для изначальной топологии свойства и основанные на них функции маршрутизации могут быть лишь дополнены, сохраняя при этом свою суть. Напомним, что под функцией маршрутизации понимают [17] формализованное правило выбора промежуточных узлов на пути от узла-источника к узлу-получателю.

2.2. РЕБЕРНОЕ МАСШТАБИРОВАНИЕ ГИПЕРКУБИЧЕСКОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Оценим возможности и особенности осуществления реберного масштабирования на примере систем с гиперкубической топологией. Эта топология достаточно широко используется и не только благодаря известным достоинствам в части простоты ее технического и алгоритмического (в частности, маршрутизации)

воплощения. Существует масса примеров того, что именно гиперкубическая информационная топология является наиболее приемлемой в распараллеливании задач [2], достаточно масштабных как по объемам обрабатываемых данных, так и по их вычислительной сложности. И если в реализуемых ранее суперкомпьютерах топология коммуникационной среды (в том числе гиперкубическая) для заказчика не имела альтернативы (как, например, в машинах семейства nCUBE [3, 4]), то предоставляемые ныне фирмами-производителями средства позволяют заказчику самому воспроизвести в своих системах интерконнект с наиболее подходящей для его задач топологией. К примеру, архитектура RSC PetaStream компании РСК «предоставляет заказчикам большую гибкость выбора типов интерконнекта и топологий создаваемой сети за счет возможности подключения различных карт расширения» [6]; суперкомпьютерная архитектура компании T-платформы также «позволяет заказчикам реализовать практически любую сетевую топологию» [16], в том числе и гиперкубическую. Добавим к этому, что заказчики суперкомпьютерных систем, как правило, изначально ориентированы на последующее поэтапное наращивание суммарной вычислительной мощности, поэтому производители снабжают первично поставляемую конфигурацию ВС избыточным числом незадействованных в ней портов или предоставляют возможности последующего дооснащения конфигурации необходимым для выбранной топологии числом сетевых адаптеров. Таким образом, возможности реберного наращивания топологии ВС необходимую конструктивную базу, как правило, имеют.

Не оценивая пока что осуществимость реберного масштабирования в гиперкубе с позиции сохранения в нем функции маршрутизации, определяющей путь от источника к получателю через сопоставление двоичных кодов их адресов, рассмотрим возможные варианты такого масштабирования.

Итак, рассмотрим реберное масштабирование s -мерного гиперкуба H_s , $s \geq 2$. Порядок n исходного гиперкуба $n(H_s) = 2^s$, диаметр d равен его степени $d(H_s) = s$. Выделим в гиперкубе H_s подкуб H_k , где $2 \leq k \leq s$. Степень k -мерного H_k куба увеличим на единицу, попарно соединив вершины, находящиеся на наибольшем в нем расстоянии k . Аналогично поступим с остальными

$2^{s-k} - 1$ такими же k -мерными подкубами, составляющими в совокупности s -мерный гиперкуб H_s . Обозначим полученный в результате такого масштабирования граф как $H_{k,s+1}$. Заметим, что число добавленных при этом ребер $R(H_{k,s+1}) - R(H_s) = |E(H_{k,s+1})| - |E(H_s)| = 2^{k-1} \cdot 2^{s-k} = 2^{s-1}$, т. е. вне зависимости от выбора базового для масштабирования гиперкубического фрагмента H_k в s -мерном гиперкубе H_s это число является неизменным. Диаметр $d(H_{k,s+1})$ полученного при этом графа $H_{k,s+1}$ определится выражением $d(H_{k,s+1}) = \lceil k/2 \rceil + s - k = s - \lfloor k/2 \rfloor$ и уменьшается с возрастанием k вплоть до $k = s$ – в этом случае диаметр и приобретает минимальное значение: $d(H_{s,s+1}) = s - \lfloor s/2 \rfloor = \lceil s/2 \rceil$.

Оценка изменения ширины бисекции $B(H_{s+1})$ такого графа от k показывает, что $2 \leq k < s \Rightarrow B(H_{s+1}) = B(H_s) = n(H_s)/2 = 2^{s-1}$, однако при $k = s$ ширина бисекции увеличивается вдвое, до равного порядку $n(H_s)$ предельного значения: $k = s \Rightarrow B(H_{s+1}) = n(H_s) = 2^s$.

Таким образом, максимальный с позиции коммуникаций эффект от реберного масштабирования s -мерного гиперкуба H_s – наибольшая ширина бисекции при наименьшем диаметре и при одинаковых дополнительных расходах на интерконнект – может быть достигнут только при $k = s$, т. е. при попарном соединении вершин с равными значению s кодовыми расстояниями между ними. Естественно поэтому, что дальнейшему рассмотрению в данной работе подлежит именно этот, обладающий максимальной коммуникационной эффективностью, случай реберного масштабирования гиперкубической ВС.

В соответствии с данным в подразделе 2.1. определением, используемые в исходной гиперкубической ВС принципы маршрутизации и их программно-алгоритмическое воплощение существенных изменений вследствие реберного наращивания претерпеть не должны. Основой маршрутизации в гиперкубических ВС является принцип двоичного кодирования узлов, заключающийся в тождественности межузловых и межкодовых расстояний, когда коды любой пары смежных узлов отличаются только

одним битом [17]. Сохранение этого свойства при реберном масштабировании гиперкубической ВС позволит использовать в ней те же принципы и программно-алгоритмические средства маршрутизации, что и в исходной гиперкубической системе. Предложенный ниже способ бинарного (двухкомпонентного) кодирования вершин реберно масштабированного гиперкуба соответствует этому требованию.

Итак, подвергнем реберному масштабированию s -мерный гиперкуб H_s . Для этого увеличим до $s + 1$ разрядность двоичного адреса его вершин, добавив к нему нулевой разряд слева, и назовем такую адресацию прямой. Введем альтернативную прямой адресацию вершин, проинвертировав все биты прямого адреса и назвав этот адрес обратным. Так как старший бит прямого адреса всегда будет нулевым, то единичное его состояние является отличительным признаком обратного адреса. Обозначения прямого X и обратного \bar{X} адресов вершины x в гиперкубе H_{s+1} различаем по отсутствию и наличию знака инверсии над ними. Из определения этих адресов ясно, что кодовое расстояние между прямым и обратным адресом одной и той же вершины – $d(X, \bar{X}) = s + 1$. Известно, что любой, произвольно взятой в s -мерном гиперкубе, вершине x может быть поставлена в соответствие единственная вершина x_s , отстоящая от нее на расстоянии $d(x-x_s) = s$. То же самое справедливо и для хемминговых расстояний между кодами адресов таких вершин: $d(X, X_s) = d(\bar{X}, \bar{X}_s) = s$. Учитывая это, получим $d(X, \bar{X}_s) = d(X, \bar{X}) - d(X, X_s) = s + 1 - s = 1$, иначе говоря, $(d(X, X_s) = d(\bar{X}, \bar{X}_s) = s) \Rightarrow (d(X, \bar{X}_s) = d(\bar{X}, X_s) = 1)$. Парно соединив вершины, для которых хэмминговы расстояния между кодами прямых (или обратных) адресов вершин в H_s равны числу измерений s , получим реберно масштабированный гиперкуб H_{s+1} , в котором смежность вершин a и b задана условием:

$$d(a - b) = 1 \Leftrightarrow (d(A, B) = 1) \vee d(A, \bar{B}) = 1.$$

Учитывая, что отношения смежности на множестве всех вершин рассматриваемого графа полностью определены отношениями соседства кодов их адресов, расстояние $d(a-b)$ между любыми двумя вершинами реберно масштабированного гиперкуба

H_{s+1} определится минимальным из кодовых расстояний между прямыми/обратными их адресами в любом их сочетании:

$$d(a, b) = \min d(A, B), d(A, \bar{B}), d(\bar{A}, B).$$

Продемонстрируем вышесказанное на примере изображенного на рис. 1 реберно масштабированного 4-мерного гиперкуба H_{4+1} . Вершины в нем сохраняют соответствующие исходному 4D-кубу H_4 пометки; для удобства перевода в двоичную кодировку дадим их в восьмеричной системе. Соответствие прямого X и обратного \bar{X} кодов адресов вершин этим пометкам дано в представленной ниже табл. 1:

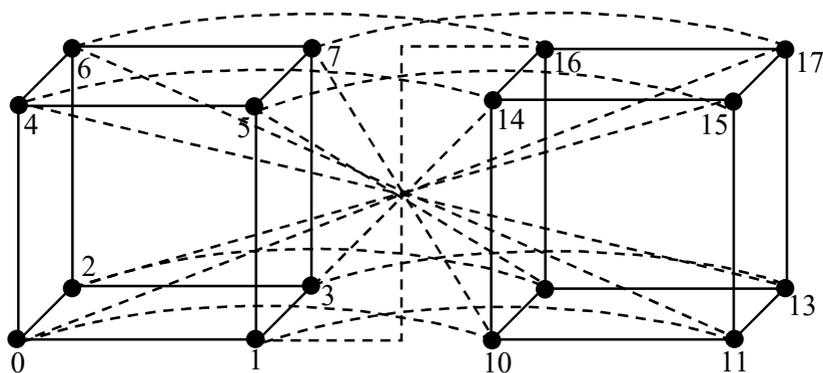


Рис. 1. Реберно масштабированный 4-мерный гиперкуб H_{4+1}

Таблица 1. Соответствие прямых X и обратных \bar{X} адресов вершин x в гиперкубе H_{4+1}

x	X	\bar{X}	x	X	\bar{X}
0	00000	11111	10	01000	10111
1	00001	11110	11	01001	10110
2	00010	11101	12	01010	10101
3	00011	11100	13	01011	10100
4	00100	11011	14	01100	10011
5	00101	11010	15	01101	10010
6	00110	11001	16	01110	10001
7	00111	11000	17	01111	10000

Для взятых в этом графе вершин $a = 5$ и $b = 16$ получим $A = 00101$, $B = 01110$ и $\bar{A} = 11010$, $\bar{B} = 10001$. Как видим, расстояние Хэмминга между отнесенными к вершине прямым и обратным кодами ее адресов не зависит от выбора вершины и равно увеличенному на единицу числу измерений s : $d(00101, 11010) = 5$ и $d(01110, 10001) = 5$. Тожественность междуузловых и межкодовых расстояний в H_{s+1} $d(a - b) = d(A, B) = d(\bar{A}, \bar{B})$ видна из $d(5-16) = d(00101, 01110) = (11010, 10001) = 3$. Коды смежных в H_4 вершин 5 и 7 ($d(5-7) = 1$) являются соседними хоть в прямой ($d(00101, 00111) = 1$), хоть в обратной ($d(11010, 11000) = 1$) их кодировке. Хэммингово расстояние между кодами адресов максимально разнесенных в гиперкубе H_4 друг от друга вершин $a = 5$, $a_4 = 12$ с расстоянием $d(5-12) = 4$ между ними тождественны этому расстоянию: $d(A, A_4) = d(00101, 01010) = 4$ и $d(\bar{A}, \bar{A}_4) = d(11010, 10101) = 4$. При этом хэммингово расстояние между прямым адресом A вершины $a = 5$ и инвертированным адресом \bar{A}_4 максимально удаленной от нее вершины $a_4 = 12$ равно единице: $d(A, \bar{A}_4) = d(00101, \bar{01010}) = d(00101, 10101) = 1$. Таким образом, соседство¹ прямого кода одной вершины и обратного кода другой указывает на смежность вершин в графе H_{4+1} . Расстояние между вершинами $a = 5$ и $b = 16$ в H_{4+1} – $d(5, 16) = \min(d(00101, 01110), d(00101, 10001)) = \min(3, 2) = 2$. В 7D-кубе H_{7+1} , например, для $a = 5$ и $b = 132$ получим: $A = 00000101$, $B = 01011010$; $\bar{A} = 11111010$, $\bar{B} = 10100101$ и $d(5, 132) = \min(d(00000101, 01011010), d(00000101, 10100101)) = \min(6, 2) = 2$.

Итак, реберное масштабирование ВС с гиперкубической топологией существенно улучшает общепринятые в сопоставлении топологий сетевые показатели:

- диаметр уменьшается вдвое – $d(H_{s+1}) = d(H_s)/2$;
- ширина бисекции увеличивается вдвое – $B(H_{s+1}) = 2B(H_s)$.

При этом, как показано выше, принцип тождественности расстояний между вершинами реберно масштабированного гиперкуба

¹ Кодовые слова с единичным расстоянием между ними называют соседними.

и кодовых расстояний между их адресами сохраняется, а следовательно, сохраняется возможность использования основанных на этом принципе алгоритмов маршрутизации.

Отметим также, что заказчики суперкомпьютерных систем, как правило, изначально ориентированы на поэтапное наращивание суммарной вычислительной мощности, поэтому производители систем с фиксированными топологиями (как, например, *иCUBE*) снабжают первично поставляемую конфигурацию ВС избыточным числом незадействованных в ней портов [3, 4]. Для систем же, топологии которых могут конфигурироваться заказчиками, такие производители, как российские РСК [6] и Т-Платформ [16], предоставляют возможность последующего дооснащения первоначально поставляемой конфигурации необходимым для создания выбранной заказчиком топологии числом сетевых адаптеров. Таким образом, и в том, и в другом случаях реберное масштабирование конструктивно вполне осуществимо и не требует характерных для традиционного масштабирования затрат на наращивание в системе числа процессоров.

3. Топологическая масштабируемость реберно модифицируемой гиперкубической ВС.

3.1. БАЗОВЫЕ ПОСТАНОВКИ И ОПРЕДЕЛЕНИЯ.

Постановка задачи распараллеливания алгоритмов с учетом допускаемой достижимости информационно связанных процессоров впервые предложена и использована автором для оценки отказоустойчивости вычислительных систем [14]. Дальнейшее развитие эта постановка получила в работе [7], где был предложен аналитический подход к синтезу регулярных графов с заданными значениями порядка, степени и обхвата, а также в [9], где была поставлена и решена задача синтеза регулярных структур ВС, обладающих при этом минимальными задержками, и в [8], где рассмотрена проблема выявления в графе вычислительной системы компонент, соответствующих размещаемой подсистеме в отношении достижимости и соединимости ее вершин и предложен аналитический метод ее решения, основанный, как и все перчисленные здесь работы, на проективном описании графа вычислительной системы [13], [12]. Развернутое обоснование,

основанное на предложенной модели параллельных вычислений, эта постановка впервые получила в [10].

В связи с тем, что полная связность всех процессоров системы при неограниченном ее масштабировании реализована быть не может, а условие физической смежности задействованных в параллельной задаче процессоров не всегда может быть реализовано даже при малом их числе, в [8] информационную модель ВС предложено представлять графами δ -достижимости, отношения смежности вершин в которых удовлетворяют допускаемым размещаемыми подсистемами требованиям в части предельно допускаемых расстояний δ между ними. Максимальное по включению подмножество вершин графа δ -достижимости G_δ , связанных между собой отношениями δ -достижимости, порождает в нем клику $K(G_\delta)$, (δ -клику $K_\delta(G)$ графа G системы), множество вершин которой объединяет в себе подмножество всех основных вершин глобальной δ -компоненты этого графа и подмножество вспомогательных вершин, входящих в состав δ -путей между основными вершинами лишь в качестве промежуточных и не принадлежащих клике $K(G_\delta)$. Подобно плотности $\varphi(G)$ графа G , определяемой порядком его наибольшей клики [8, с. 21], порядок наибольшей из максимальных δ -клик графа $G(V, E)$ называем плотностью $\varphi(G_\delta)$ графа δ -достижимости, или δ -плотностью $\varphi_\delta(G)$ графа $G(V, E)$ — $\varphi(G_\delta) \equiv \varphi_\delta(G)$.

Пусть проблемная ориентация гиперкубической ВС с используемой ею NT определена перечнем распараллеливаемых пользовательских приложений, каждое из которых отвечает условиям неограниченной распараллеливаемости [1], т. е. не содержит скалярных фрагментов и характеризуется объемами (или диапазонами изменения) W и Q вычислительных и обменных операций, равномерно распределяемыми между процессорами подсистемы в зависимости от числа параллельных ветвей p . Согласно предложенной в [10] модели существуют предельные значения допускаемых для каждой из этих задач достижимостей

∂ , и все решаемые в исследуемой ВС пользовательские задачи могут быть классифицированы в соответствии с этими значениями ∂ . Тогда выбор топологии, адекватной перечню решаемых в исследуемой системе задач может быть обоснован анализом функций ∂ -плотности сопоставляемых топологий на соответствующем этим задачам диапазоне $\partial_{\min} \leq \partial \leq \partial_{\max}$ изменения достижимостей.

В работе [11] проведено исследование и определена формальная зависимость порядка $n_{\partial}(H_s)$ максимальной по включению ∂ -компоненты s -мерного гиперкуба H_s , или, что равнозначно, ∂ -плотности $\varphi_{\partial}(H_s) \equiv n_{\partial}(H_s)$, от предельно допускаемых задачами расстояний $\partial \leq s$ между информационно-смежными в этих задачах процессорами:

$$(1) \quad \varphi_{\partial}(H_s) = \begin{cases} \sum_{i=0}^{\partial/2} \binom{s}{i} & \text{четн. } \partial, \\ \binom{s-1}{\lfloor \partial/2 \rfloor} + \sum_{i=0}^{\lfloor \partial/2 \rfloor} \binom{s}{i} & \text{нечетн. } \partial. \end{cases}$$

В следующем подразделе получим такую зависимость для реберно масштабированного гиперкуба H_{s+1} .

3.2. АНАЛИЗ ∂ -ПЛОТНОСТИ РЕБЕРНО МОДИФИЦИРОВАННОГО ГИПЕРКУБА.

Исследование формальной зависимости ∂ -плотности $\varphi_{\partial}(H_{s+1})$ осуществим исходя из анализа расстояний между кодами двоичных адресов вершин исходного гиперкуба H_s . Как известно, расстояние $d(a-b)$ между произвольно взятыми в гиперкубе вершинами $a \neq b$ равно кодовому расстоянию $d(A, B)$ между двоичными кодами адресов A, B этих вершин. Для удобства изложения далее мы сохраняем соответствие восьмеричной нумерации вершин двоичному кодированию их прямых адресов (с добавленным нулевым старшим разрядом).

Итак, вершины H_{s+1} , как и вершины исходного гиперкуба $H_s \subset H_{s+1}$, являющегося остовным¹ подграфом H_{s+1} , пронумерованы в восьмеричной системе от 0 до $2^s - 1$, и смежным в H_s вершинам соответствуют соседние в двоичной кодировке прямые в H_{s+1} адреса, причем нулевой вершине a_0 соответствуют нулевой

¹ Множества вершин этих графов совпадают

прямой адрес A_0 , все $s + 1$ разрядов которого – нулевые (вес его двоичного кода $- |A_0| = 0$), и обратный адрес \bar{A}_0 , все $s + 1$ разрядов которого – единичные, и вес его двоичного кода $- |\bar{A}_0| = s + 1$.

Как известно, кодовое расстояние $d(A, B)$ между двоичными кодами адресов A и B вершин a и b определяется их поразрядным сложением по модулю 2: $d(A, B) = |A \oplus B|$.

Из свойства поразрядного сложения, согласно которому суммирование разряда c с единичным инвертирует значение этого разряда $- c \oplus 1 = \bar{c}$, получим: $B \oplus \bar{A}_0 = \bar{B}$, $A \oplus \bar{B} = A \oplus B \oplus \bar{A}_0 = A \oplus \bar{B}$. Тогда $|A \oplus \bar{B}| = s + 1 - |A \oplus B|$, и расстояние между вершинами a, b в графе H_{s+1} , полученном в результате реберной надстройки гиперкуба H_s , определится из:

$$d(a - b) = \min\{d(A, B), s + 1 - d(A, B)\}.$$

Отсюда понятно, что расстояние между двумя произвольными вершинами a, b графа H_{s+1} не превышает половины от $s + 1$ и определено на интервале от 1 до $\lfloor (s + 1)/2 \rfloor$:

$$1 \leq d(a - b) \leq \lfloor (s + 1)/2 \rfloor,$$

что соответствует высказанному выше утверждению об уменьшенном диаметре графа H_{s+1} в сравнении с диаметром $d(H_s) = s$.

Произведем оценку хэмминговых расстояний между кодами прямых адресов произвольной пары вершин $a_i \neq a_j$ графа H_{s+1} , где индексы i, j при вершинах a_i, a_j соответствуют весам $|A_i|, |A_j|$ двоичных кодов A_i, A_j прямых адресов этих вершин и указывают на их кодовые расстояния $d(A_0, A_i), d(A_0, A_j)$ от адреса A_0 ракурсной вершины a_0 : $i = |A_i| = d(A_0, A_i)$, $j = |A_j| = d(A_0, A_j)$. Множества вершин с соответствующим индексам i, j числом единичных разрядов обозначим через $A(i), A(j)$: $A(i) = \{a_i \mid i = |A_i|\}$, $A(j) = \{a_j \mid j = |A_j|\}$.

Пусть $0 < i \leq j \leq \lfloor (s + 1)/2 \rfloor$, и $x_{(1)} \leq i$ единичных разрядов A_i совпадают с разрядами A_j . Тогда кодовое расстояние между A_i и A_j в зависимости от $x_{(1)}$ составит:

$$(2) \quad d(A_i, A_j) = (i - x_{(1)}) + (j - x_{(1)}) = (i + j) - 2x_{(1)}.$$

В соответствии с (2), минимальным на множестве пар вершин $\{a_i \in A(i), a_j \in A(j)\}$ кодовое расстояние будет при $x_{(1)} = i$: $d_{\min}(A_i, A_j) = (i + j) - 2i = j - i$, а максимальное кодовое расстояние соответствует максимальному несовпадению ($x_{(1)} = 0$) этих адресов, при этом $d_{\max}(A_i, A_j) = i + j$.

Учитывая, что $d(A, \bar{B}) = s + 1 - d(A, B)$, из (2) получим:
 $d(A_i, \bar{A}_j) = s + 1 - d(A_i, B_j) = (s + 1) - (i + j) + 2x_{(1)}$.

Так как расстояние $d(a_i - a_j)$ между вершинами $a_i \in A(i)$, $a_j \in A(j)$ в графе H_{s+1} определяется минимумом из расстояний $d(A_i, A_j)$ и $s + 1 - d(A_i, A_j)$, определим предельные значения i, j , при которых попарные расстояние между вершинами из соответствующих множеств $A(i), A(j)$ не превысят заданного значения достижимости ∂ . Это соответствует условиям:

$$(3) \quad (i + j'(\partial)) - 2x_{(1)} \leq \partial,$$

или

$$(4) \quad (s + 1) - (i + j''(\partial)) + 2x_{(1)} \leq \partial.$$

Множество $V_\partial(H_{s+1})$ взаимно ∂ -достижимых в графе H_{s+1} вершин состоит из двух соответствующих неравенствам (3), (4) подмножеств: из заданного в (3) подмножества $V'_\partial(H_{s+1})$ вершин, расстояния между которыми определены кодовыми расстояниями между их прямыми адресами, и из заданного в (4) подмножества вершин $V''_\partial(H_{s+1})$, расстояния которых от вершин из $V'_\partial(H_{s+1})$ определены кодовыми расстояниями обратных адресов вершин из $V''_\partial(H_{s+1})$ от прямых адресов вершин из $V'(\partial)$ при том, что расстояния между вершинами внутри $V''_\partial(H_{s+1})$ равны кодовым расстояниям между их прямыми или обратными адресами. Учитывая существенное отличие кодовых расстояний $j = |A_j|$ прямых адресов вершин из подмножеств $V'_\partial(H_{s+1})$ и из $V''_\partial(H_{s+1})$ от A_0 , значения j в (3), (4) проиндексированы соответственно. Понятно, что на интервале $j'(\partial) < j < j''(\partial)$ расстояния $d(a_i - a_j) > \partial$, поэтому ни одна вершина подмножеств $V(j) | j'(\partial) < j < j''(\partial)$ не может войти ни в $V'_\partial(H_{s+1})$, ни в $V''_\partial(H_{s+1})$ и, соответственно, не может быть включена в число вершин, определяющих искомым нами порядок ∂ -клик (∂ -плотность $\varphi_\partial(H_{s+1})$).

Из (3) имеем $i + j'(\partial) \leq \partial + 2x_{(1)}$. Так как это неравенство должно быть справедливым для любого $i \leq j'(\partial)$, то подставив максимальное значение $i = j'(\partial)$, получим $j'(\partial) \leq \frac{\partial + 2x_{(1)}}{2}$. Понятно, что минимальным значение $j'(\partial)$ будет при максимальном несовпадении разрядов в прямых адресах A_i, A_j вершин $a_i \in A(i)$,

$a_j \in A(j'(\partial))$, что соответствует $x_{(1)} = 0$, поэтому окончательно рассматриваемое условие при $\partial < \lfloor (s+1)/2 \rfloor^1$ имеет вид:

$$(5) \quad j'(\partial) \leq \lfloor \partial/2 \rfloor < \lfloor (s+1)/4 \rfloor.$$

В соответствии с (5) и с формулой ∂ -плотности гиперкуба H_s (1) получим мощность подмножества $V'_\partial(H_{s+1})$:

$$|V'_\partial(H_{s+1})| = \begin{cases} \sum_{i=0}^{\partial/2} \binom{s}{i} & \text{четн. } \partial, \\ \binom{s-1}{\lfloor \partial/2 \rfloor} + \sum_{i=0}^{\lfloor \partial/2 \rfloor} \binom{s}{i} & \text{нечетн. } \partial. \end{cases}$$

В отличие от (3) условие (4) отнесено к вершинам, коды прямых адресов A_j которых содержат большее, чем $\lfloor \partial/2 \rfloor$ число единичных разрядов ($|A_j| > \lfloor \partial/2 \rfloor$), и область допустимых при заданной достижимости ∂ значений $j_2(\partial) < j \leq s$ будет отличной от $0 < j \leq j'(\partial)$. Учитывая, что максимум полученного из (4) $j'' \geq s+1-i+2x_{(1)}-\partial$ соответствует $x_{(1)}=i$ и максимальному из $i \leq j'(\partial)$ значению $i=j'(\partial)=\lfloor \partial/2 \rfloor$, получим $j'' \geq s+1-\lfloor \partial/2 \rfloor$. Однако, исходя из того, что $d(A_0, \bar{A}_0) = 1$ получим $j'' \geq s - \lfloor \partial/2 \rfloor = s+1-\lfloor \partial/2 \rfloor$. Из условий $j'' \geq s+1-\lfloor \partial/2 \rfloor$ и $j'' \geq s+1-\lfloor \partial/2 \rfloor$ выбираем более жесткое: $j'' \geq s+1-\lfloor \partial/2 \rfloor$.

Тогда для $|V''_\partial(H_{s+1})|$ имеем:

$$|V''_\partial(H_{s+1})| = \begin{cases} \sum_{i=0}^{\partial/2-1} \binom{s}{i} & \text{четн. } \partial, \\ \binom{s-1}{\lfloor \partial/2 \rfloor - 1} + \sum_{i=0}^{\lfloor \partial/2 \rfloor - 1} \binom{s}{i} & \text{нечетн. } \partial. \end{cases}$$

Сумма $|V'_\partial(H_{s+1})|$ и $|V''_\partial(H_{s+1})|$ для анализируемого реберно масштабированного гиперкуба H_{s+1} определяет порядок компоненты ∂ -достижимости этого графа, или искомую его ∂ -плотность:

$$\varphi_\partial(H_{s+1}) = |V_\partial(H_{s+1})| = |V'_\partial(H_{s+1})| + |V''_\partial(H_{s+1})|.$$

Исходя из этого для четных значений ∂ получим:

¹ Выше мы показали, что при $\partial \geq \lfloor (s+1)/2 \rfloor$ все вершины H_{s+1} входят в состав его ∂ -клик, и его ∂ -плотность $\varphi_\partial(H_{s+1}) = 2^s$.

$$\varphi_{\partial}(H_{s+1}) = \sum_{i=0}^{\partial/2} \binom{s}{i} + \sum_{i=0}^{\partial/2-1} \binom{s}{i},$$

а для нечетных –

$$\varphi_{\partial}(H_{s+1}) = \binom{s-1}{\lfloor \partial/2 \rfloor} + \sum_{i=0}^{\lfloor \partial/2 \rfloor} \binom{s}{i} + \binom{s-1}{\lfloor \partial/2 \rfloor - 1} + \sum_{i=0}^{\lfloor \partial/2 \rfloor - 1} \binom{s}{i}.$$

Используя общеизвестное для биномиальных коэффициентов соотношение

$$\binom{s+1}{i} = \binom{s}{i-1} + \binom{s}{i},$$

для четных ∂ получим:

$$\sum_{i=0}^{\partial/2} \binom{s}{i} + \sum_{i=0}^{\partial/2} \binom{s}{i-1} = \sum_{i=0}^{\partial/2} \binom{s+1}{i}.$$

Выполнив аналогичные действия для нечетных значений достижимости ∂ , получим результирующую формулу ∂ -плотности реберно масштабированного графа H_{s+1} при $\partial < \lfloor (s+1)/2 \rfloor$ и $s > 2$; в иных случаях – при $\partial \geq \lfloor (s+1)/2 \rfloor$ $\varphi_{\partial}(H_{s+1}) = 2^s$, а при $s = 2$ $\varphi_1(H_{2+1}) = 4$:

$$(6) \quad \varphi_{\partial}(H_{s+1}) = \begin{cases} \sum_{i=0}^{\partial/2} \binom{s+1}{i} & \text{четн. } \partial, \\ \left(\binom{s}{\lfloor \partial/2 \rfloor} + \sum_{i=0}^{\lfloor \partial/2 \rfloor} \binom{s+1}{i} \right) & \text{нечетн. } \partial. \end{cases}$$

Таким образом, ∂ -плотность реберно масштабированного гиперкуба определяется так же, как и ∂ -плотность исходного гиперкуба, но с увеличенным на единицу значением степени s . При этом интервал значений достижимости $0 < \partial < \lfloor (s+1)/2 \rfloor$ в реберно масштабированном гиперкубе в сравнении с интервалом $0 < \partial < s$ в исходном гиперкубе существенно сужен. На представленных в следующем разделе графических примерах будет показано, что это означает, что если для гиперкуба H_s линейный рост производительности системы может быть обеспечен только на задачах с достижимостью $\partial \geq s$, то после его реберной модификации к подмножеству линейно ускоряемых из набора решаемых в данной системе задач добавляются те из них, для которых $\lceil s/2 \rceil \leq \partial < s$, и таким образом число высокоэффективно решаемых в данной системе задач существенно возрастает.

3.3. МАСШТАБИРУЕМОСТЬ РЕБЕРНО МОДИФИЦИРОВАННОГО ГИПЕРКУБА.

Как мы уже говорили ранее, каждая из решаемых в исследуемой ВС (W, Q) -задач с учетом параметров используемой в системе NT может быть классифицирована по значениям достижимости $\delta(p)$, обеспечивающими при распараллеливании этой задачи на заданное число p процессоров системы требуемые значения ускорения S_p и/или эффективности E_p [15]. Поэтому далее мы считаем, что набор задач, решаемых в рассматриваемой системе с определенной сетевой технологией, распределен по значениям предельно допустимой для них достижимости $\delta \geq 1$.

В [10], [15] показано, что при решении набора задач усредненное изменение пределов их распараллеливания более выигрышно в системах с большими значениями δ -плотности их графов. Поэтому вполне оправданным является использование δ -плотности в качественной (больше/меньше – лучше/хуже) характеристике графов ВС и в сопоставительном их анализе не только для информационно полносвязных, но и для других классифицированных по значению δ задач из набора, кроме тех из них, информационные топологии которых адекватны топологии системы. Таковыми для гиперкубических ВС, например, являются задачи с кольцевой, линейной и гиперкубической топологиями. Топологически адекватные системе задачи могут быть распараллелены на все $n(G)$ ее процессоров даже при $\delta = 1$, а системы, ориентированные на решение таких задач являются специализированными. Топологическая специализация системы под некоторый достаточно ограниченный набор решаемых задач является одним из способов повышения эффективности суммарного использования всех ее процессоров, однако она (специализация) ограничивает число потенциальных пользователей системы. Анализ представленных ниже графиков и основанные на нем выводы имеют целью топологическую характеристику гиперкубической ВС, как системы универсального назначения.

Анализ функций δ -плотности гиперкуба приведен в [15], поэтому на характерных особенностях этих функций останавливаться здесь мы не будем и дадим графическое их сопоставление

(см. рис. 2) с функциями δ -плотности предложенного нами реберно масштабированного гиперкуба. Из этого рисунка ясно, что с ростом числа измерений s реберно масштабированного гиперкуба функция δ -плотности растет линейно ($\varphi_\delta(H_{s+1}) = 2^s$) пока $s \leq 2\delta$. Последующее наращивание $s \geq 2\delta + 1$ приводит к резкому падению δ -плотности реберно модифицированного гиперкуба от значения $\varphi_\delta(H_{s+1}) = 2^s$ при $s = 2\delta$ до значения δ -плотности $\varphi_\delta(H_{s+1}) <$, характерного для масштабированного гиперкуба с

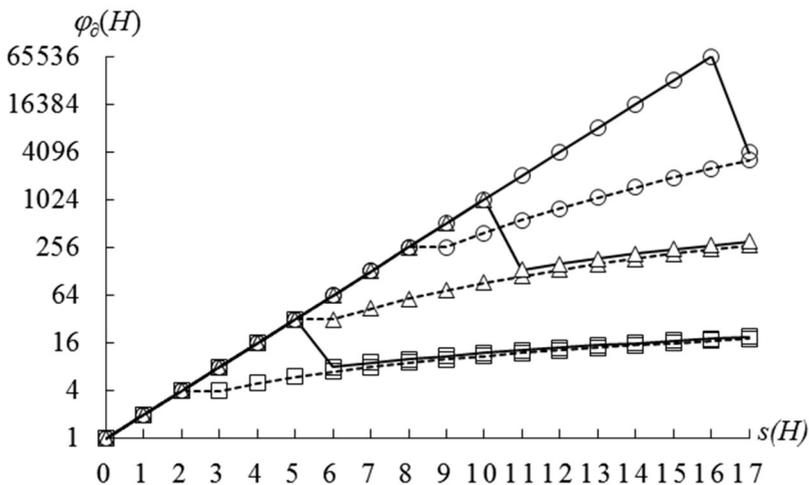


Рис. 2. Функции δ -плотности $\varphi_\delta(H)$:
 гиперкуба H_s (штриховые линии) и реберно
 масштабированного гиперкуба H_{s+1} (сплошные линии);
 □ — $\delta = 2$, △ — $\delta = 5$, ○ — $\delta = 8$.

числом измерений, превышающим значение аргумента s на единицу: $s > 2\delta \Rightarrow \varphi_\delta(H_{s+1}) = \varphi_\delta(H_{(s+1)})$; здесь в отличие от H_{s+1} ограниченный скобками индекс — $H_{(s+1)}$ определяет обычный гиперкуб с числом измерений, равным $(s + 1)$. Это означает, что несмотря на падение δ -плотности реберно масштабированного гиперкуба при $s \geq 2\delta$ ее значение все равно будет большим, чем для обычного s -куба на величину, соответствующую увеличенному на единицу значению s .

В приведенную ниже таблицу 2 сведена часть исходных для

построения рисунка 2 расчетных данных, достаточная лишь для численной иллюстрации вышесказанного. Например, в гиперкубе H_9 с числом процессоров $n(H_9) = 512$ при $\delta = 5$ имеем плотность $\varphi_5(H_{10}) = 92$ – заметим, что этот случай с $s = 9$, $\delta = 5$ соответствует условию $\delta \geq \lceil s/2 \rceil$. Масштабирование гиперкуба H_9 до H_{10} увеличивает число процессоров вдвое до $n(H_{10}) = 1024$, но повышает его δ -плотность всего лишь до $\varphi_5(H_{10}) = 112$, тогда как реберное модифицирование этого же гиперкуба до H_{9+1} , не требуя при этом увеличения числа процессоров – $n(H_{9+1}) = n(H_9) = 512$, повышает δ -плотность до максимально возможного значения $\varphi_5(H_{9+1}) = n(H_9) = 512$. В случае с $\delta = 4$ $s = 9$, когда условие $\delta \geq \lceil s/2 \rceil$ не выполняется, масштабирование числа процессоров с $n(H_9) = 512$ до $n(H_{10}) = 1024$ повышает плотность с $\varphi_4(H_9) = 46$ до $\varphi_4(H_{10}) = 56$, тогда как реберное масштабирование $H_9 \rightarrow H_{9+1}$ повышает плотность до этого же значения $\varphi_4(H_{9+1}) = \varphi_4(H_{10}) = 56$ без наращивания числа процессоров в системе – увеличением только числа ребер. Аналогичную картину можно увидеть из таблицы 2 для задач с $\delta = 2 \leq \lceil s/2 \rceil$ при масштабировании гиперкуба H_{10} : $\varphi_2(H_{10}) = 11$, $\varphi_2(H_{11}) = 12$, $\varphi_2(H_{10+1}) = \varphi_2(H_{11}) = 12$.

Таблица 2. Значения функций δ -плотности: гиперкуба – $\varphi_\delta(H_s)$ и реберно масштабированного гиперкуба – H_{s+1}

$\varphi_\delta(H)$	H_s			H_{s+1}		
$\delta \backslash s$	9	10	11	9 + 1	10 + 1	11 + 1
1	2	2	2	2	2	2
2	10	11	12	11	12	13
3	18	20	22	20	22	24
4	46	56	67	56	67	79
5	74	92	112	512	1024	134
6	130	176	232	512	1024	2048
7	186	260	352	512	1024	2048
8	256	386	562	512	1024	2048
9	512	512	772	512	1024	2048
10	512	1024	1024	512	1024	2048

Таким образом, если набор решаемых в s -мерной гиперкубической ВС задач определен соответствующим используемой NT интервалом изменения достижимости $\partial_{\min} \leq \partial \leq \partial_{\max}$, то при реберном масштабировании этой ВС часть задач набора, для которых $\lceil s/2 \rceil \leq \partial \leq \partial_{\max}$, в потенциале покажут линейный рост производительности вплоть до максимального их распараллеливания на $p = 2^s$ процессоров. Заметим, что в традиционном гиперкубе такой рост мог бы быть обеспечен лишь на задачах с $s \leq \partial$, что существенно сужает число таких задач и предъявляет значительно более жесткие требования к производительности используемой в системе NT .

Итак, несмотря на то, что в наилучшем случае распараллеливания задач с $\partial_{\min} \leq \partial < \lceil s/2 \rceil$, линейного роста эффективности при реберном масштабировании гиперкубической ВС добиться не удастся, потенциал эффективного распараллеливания задач в такой ВС все же превышает потенциал обычной гиперкубической ВС: такого же в численном выражении эффекта можно достигнуть увеличением числа ее процессоров вдвое, т. е. при пропорционально большей цене. Чтобы убедиться в этом, сопоставим стоимости реберного и традиционного масштабирования s -мерной гиперкубической ВС.

Обычное масштабирование наряду с двойным увеличением числа процессоров от 2^s до $2^{(s+1)}$ требует более чем двойного увеличения числа ребер E в графе $H_{(s+1)}$ от $E(H_s) = 2^s \cdot s/2 = s \cdot 2^{s-1}$ до $E(H_{(s+1)}) = 2^{(s+1)} \cdot (s+1)/2 = (s+1) \cdot 2^s$. Реберное масштабирование графа H_s при неизменном числе процессоров в нем увеличивает число ребер от $E(H_s) = s \cdot 2^s$ до $E(H_{s+1}) = 2^s \cdot (s+1)/2 = (s+1) \cdot 2^{s+1}$, т. е. всего в $(1 + s^{-1})$ раза.

Таким образом понятно, что даже в наилучшем случае использования недостаточно быстродействующей для набора решаемых задач NT , получение одинакового ускорения в гиперкубической ВС достигается практически вдвое дешевле реберным ее масштабированием, чем традиционным. Для задач же с $\partial \geq \lceil s/2 \rceil$ потенциал распараллеливания в реберно масштабированной гиперкубической ВС максимален и ограничен лишь числом имеющихся в ее составе процессоров.

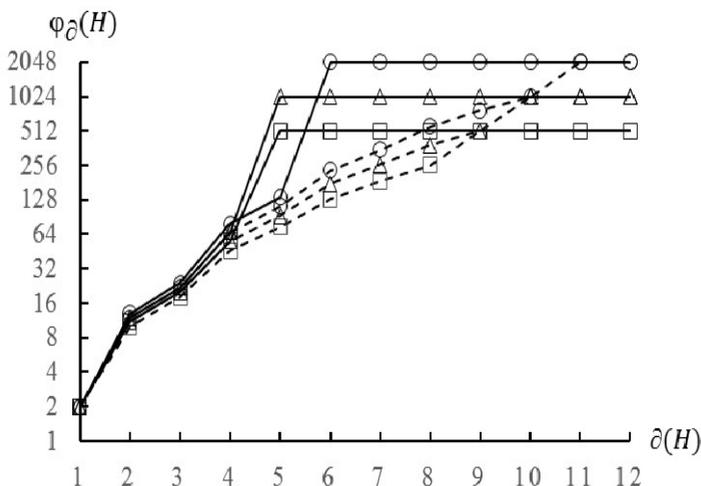


Рис. 3. Функции δ -плотности $\varphi_{\delta}(H)$: гиперкуба H_s (штриховые линии) и реберно масштабированного гиперкуба H_{s+1} (сплошные линии); $\square - s = 9$, $\Delta - s = 10$, $\circ - s = 11$.

В том, что реберное, в сравнении с традиционным, масштабирование гиперкуба позволяет добиться существенно большего приращения числа эффективно используемых в решении данной задачи процессоров при условии $\lceil s/2 \rceil \leq \delta < s$, можно убедиться и из рисунка 3, сопоставляющем графики функций плотности гиперкубов H_9 , H_{10} и H_{11} и их реберно масштабированных модификаций H_{9+1} , H_{10+1} и H_{11+1} , отнесенные к параллельным задачам, классифицированным по значению предельно допускаемой между их ветвями достижимости δ .

Функция топологической масштабируемости $\mu_{\delta}(G)$ системы определена в [10] нормированной порядком $n(G)$ ее графа δ -плотностью $\varphi_{\delta}(G)$: $\mu_{\delta}(G) = \varphi_{\delta}(G)/n(G)$. Учитывая, что предельные значения достижимостей δ в системе обусловлены предписанными ей задачами и используемой сетевой технологией, функция топологической масштабируемости системы характеризует изменение потенциала в распараллеливании предписанных ей задач при изменении порядка n графа G_n системы (в нашем случае $G_n = H_s$

и $n(H_s) = 2^s$, поэтому в качестве аргумента удобней использовать число измерений s исходного гиперкуба H_s). Приведенный ниже рисунок 4 позволяет графически сопоставить функции топологической масштабируемости исходного $\mu_{\partial}(H_s)$ и реберно масштабированного гиперкуба $\mu_{\partial}(H_{s+1})$ на классифицированных по значениям ∂ задачах.

Из рисунка 4 видно, что при традиционном наращивании гиперкубической ВС функция ее топологической масштабируемости на задачах ∂ -класса имеет максимальное, равное единице, значение, пока $s \leq \partial$. Увеличение числа измерений гиперкуба на

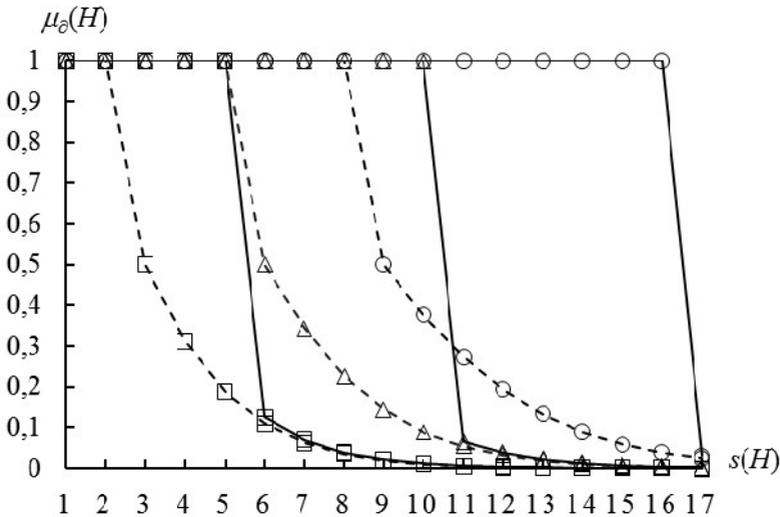


Рис. 4. Функции масштабируемости $\mu_{\partial}(H)$:

$\mu_{\partial}(H_s)$ – гиперкуб (штриховые линии),

$\mu_{\partial}(H_{s+1})$ – реберно масштаб. гиперкуб (сплошные линии).

□ – $\partial = 2$, Δ – $\partial = 5$, \circ – $\partial = 8$.

единицу сверх ∂ до $s = \partial + 1$ удваивает число процессоров в системе, но предел распараллеливания ∂ -задач при этом сохраняется на прежнем уровне, и это соответствует вдвое меньшему значению функции топологической масштабируемости: $\mu_{\partial}(H_{(\partial+1)}) = 0,5$. Последующее увеличение s до значений $s > \partial + x$,

$x > 0$ приводит к незначительному, в сравнении с увеличивающимся при этом числом процессоров, росту функции δ -плотности $\varphi_{\delta}(H_s)$ и, соответственно, к монотонному убыванию функции топологической масштабируемости $\mu_{\delta}(H_s)$:

$$(s = \delta + x, x \geq 0) \Rightarrow \mu_{\delta}(H_{(\delta+x+1)}) < \mu_{\delta}(H_{(\delta+x)}).$$

Сочетание традиционного масштабирования гиперкубической ВС с реберным ее масштабированием сохраняет равное единице значение функции топологической масштабируемости уже до $s = 2\delta$, тем самым в два раза расширяя границы эффективного масштабирования задач с соответствующим используемой в системе NT значением предельной достижимости δ . Несмотря на то, что при последующем ($s > 2\delta + x, x > 0$) масштабировании гиперкубической ВС происходит резкое падение функций ее топологической масштабируемости, значение функции $\mu_{\delta}(H_{(2\delta+x+1)})$ реберно масштабированной ВС со степенью $s = 2\delta + x$ все равно будет превышать значение $\mu_{\delta}(H_{(2\delta+x)})$:

$$(s = 2\delta + x, x \geq 0) \Rightarrow \mu_{\delta}(H_{(2\delta+x+1)}) > \mu_{\delta}(H_{(2\delta+x)}).$$

4. Заключение

Наращивание числа процессорных элементов преследует, как правило, одну из или сразу обе цели: предотвращение коллизий, возникающих в связи с дефицитным суммарным потенциалом текущей конфигурации ВС, недостаточным для удовлетворения совмещенных во времени возрастающих потребностей множества пользователей и их задач, и/или увеличение потенциала масштабирования задач для повышения оперативности их решения в данной системе. В работе предложен и исследован названный реберным масштабированием способ достижения второй из этих целей без дорогостоящего увеличения числа процессорных элементов: наращиванием числа ребер исходной топологии, не требующим перекоммутации имеющихся межпроцессорных связей и изменения соответствующих исходной топологии системы способов адресации и маршрутизации.

Даны общее описание предложенного способа и его конкретизация для вычислительных систем с гиперкубической топологией. Показано, что традиционно используемые в сопоставлении коммуникационных возможностей сетей связи показатели – ширина бисекции сети и ее диаметр, в сравнении гиперкубической топологии с ее реберно модифицированным вариантом свидетельствуют об удвоенном преимуществе последнего. Это побудило к поиску возможности сохранения в этом варианте имеющих известные преимущества способов адресации и маршрутизации, свойственных исходной гиперкубической топологии. Исследовано также влияние предложенных в работе изменений гиперкубической ВС на потенциалы распараллеливаемости в ней задач, классифицированных по лимитированной в них достижимости.

Традиционная для гиперкуба адресация вершин, в которой расстояния между его вершинами равны кодовым расстояниям между их адресами, для реберно масштабированного гиперкуба дополнена инверсной (обратной) адресацией, основанной на этом же принципе. Показано, что увеличение при этом разрядности адресного слова на единицу позволяет дополнить гиперкуб ребрами между наиболее удаленными в нем вершинами, получив тем самым единичное расстояние между такими вершинами, соответствующее единичному кодовому расстоянию между прямым адресом одной из них и обратным адресом другой. Это позволяет использовать в модифицированной таким образом системе те же алгоритмы маршрутизации, что и в исходной.

Получена формула δ -плотности реберно масштабированного гиперкуба, показано, что ее значение определяется так же, как и δ -плотность исходного гиперкуба, но с увеличенным на единицу значением его степени s . Это указывает на то, что на задачах, лимитированных достижимостью $0 < \delta < \lfloor (s + 1)/2 \rfloor$, эффект от реберного масштабирования (без удвоения числа процессоров) равнозначен эффекту, который был бы получен при увеличении s на единицу (при удвоении числа процессоров). Кроме того, в реберно масштабированном гиперкубе интервал $0 < \delta < \lfloor (s + 1)/2 \rfloor$ значений лимитируемых задачами достижи-

мостей δ существенно уже интервала $0 < \delta < s$ в исходном гиперкубе. Это означает, что в отличие от исходной гиперкубической ВС, где предельному распараллеливанию вплоть до $n = 2^s$ могут быть подвергнуты только задачи с $\delta \geq s$, в реберно масштабированной ВС круг предельно распараллеливаемых задач может быть существенно расширен за счет задач с вдвое меньшими значениями предельно допускаемой достижимости $\delta \geq \lceil s/2 \rceil$. Этим же интервалам достижимости δ соответствуют и области эффективного масштабирования и реберного масштабирования гиперкубических ВС, что подтверждено в данной работе соответствующими графиками функций их δ -плотности и масштабируемости.

Следует заметить, что, в принципе, реберному масштабированию может быть подвергнут не весь s -мерный гиперкуб, а его k -мерные ($k < s$) фрагменты. Это, как показано в подразделе 2.2, несколько увеличивает диаметр результирующего графа и уменьшает ширину его бисекции. Однако очевидно, что при использовании в ВС недостаточно быстродействующей NT , обуславливающей для решаемых в системе задач малые значения допускаемой достижимости, уменьшение k может привести к повышению значений функции δ -плотности и функции масштабируемости системы именно для таких δ , а следовательно, к увеличению потенциала распараллеливания системы на задачах с малыми значениями предельно допускаемой достижимости. Вероятно, в случаях, когда набор решаемых в некоторой системе задач и используемая в ней сетевая технология имеют достаточно широкий разброс предельных достижимостей имеет смысл использовать для реберного масштабирования коммутатор, меняющий k в соответствии с потребностями (здесь речь идет о коммутации только дополняющих гиперкубическую ВС линках). Очевидно, что уменьшение при этом размера реберно масштабируемого k -куба положительно скажется и на отказоустойчивости системы, увеличивая число альтернативных маршрутов между вершинами. Однако это уже выходит за рамки поставленных в данной работе целей и требует самостоятельной проработки.

Литература

1. ВОЕВОДИН В.В., ВОЕВОДИН В.В. *Параллельные вычисления*. – Санкт-Петербург. 2002. – 599 с.
2. Гергель В.П. *Высокопроизводительные вычисления для многоядерных многопроцессорных систем*. Учебное пособие – Нижний Новгород; Изд-во ННГУ им. Н.И. Лобачевского, 2010. 421 с.
3. ДУБОВА Н. *Суперкомпьютеры nCube* // Открытые системы, № 2, 1995, С. 42-47.
4. КУЗЬМИНСКИЙ М. *Гиперкубические метаморфозы* // Computerworld Россия, № 17-18, 1997.
5. Макагон Д., СЫРОМЯТНИКОВ Е. *Сети для суперкомпьютеров* // Открытые системы, 2011, № 7, С. 1-8.
6. *Массивно-параллельная архитектура RSC PetaStream* [Электронный ресурс]. // РСК: сайт. – URL: <http://www.rscgroup.ru/ru/our-solutions> (дата обращения: 25.05.2016).
7. МЕЛЕНТЬЕВ В. А. Аналитический подход к синтезу регулярных графов с заданными значениями порядка, степени и обхвата, ПДМ, 2010, № 2, 74–86
8. МЕЛЕНТЬЕВ В. А. *Вложение подсистем, лимитирующих длину и число путей между вершинами графа вычислительной системы* / УБС. 47. М.: ИПУ РАН, 2014. С.212-246.
9. В. А. Мелентьев *Компактные структуры вычислительных систем и их синтез*, УБС, 32 (2011), 241–261
10. МЕЛЕНТЬЕВ В. А. *О топологической масштабируемости вычислительных систем* / УБС. 58. М.: ИПУ РАН, 2015. С.115-143.
11. МЕЛЕНТЬЕВ В. А. *Предельное конфигурирование подсистем в гиперкубических вычислительных системах*. // Информационные технологии и вычислительные системы. 2015. № 2. С. 20-30.
12. МЕЛЕНТЬЕВ В. А. *Скобочная форма описания графов и ее использование в структурных исследованиях живучих вычислительных систем* //Автометрия. – 2000. – Т. 38. – №. 4. – С. 36-52.

13. МЕЛЕНТЬЕВ В. А. *Формальные основы скобочных образов в теории графов* // Труды Второй Междунар. конф. Параллельные вычисления и задачи управления РАСО. – 2004. – С. 694-706.
14. В. А. МЕЛЕНТЬЕВ *Функция структурной отказоустойчивости и δ -ограниченная компонента связности графа вычислительной системы*, ПДМ, 2008, № 2, 102–106
15. МЕЛЕНТЬЕВ В.А., ШУБИН В.И., ЗАДОРОВНИКОВ А.Ф. *Топологическая масштабируемость гиперкубических параллельных систем и задач* // ISJ Theoretical & Applied Science 11 (31): 122-129.
16. *T-Платформы. Суперкомпьютерные решения* [Электронный ресурс]. // t-ПЛАТФОРМЫ: сайт. – URL: <http://www.t-platforms.ru/products/hpc/v-class.html> (дата обращения: 25.05.2016).
17. ЦИЛЬКЕР Б.Я. *Организация ЭВМ и систем: учебник для вузов* / С.А. Орлов, Б.Я. Цилькер. - СПб.: Питер, 2011. - 688 с.
18. A AMDAHL GM (1967) Validity of the single-processor approach to achieving large-scale computing capabilities // Proc. AFIPS Conference, Atlantic City, N.J., Apr. 18-20. – AFIPS Press, Reston, Va. – 1967, Vol. 30, pp. 483-485.

THE "EDGE" SCALING OF THE HYPERCUBIC COMPUTING SYSTEMS

Victor Melentiev, Rzhanov Institute of Semiconductor Physics Siberian Branch of RAS, Novosibirsk, Cand. Sc., senior research associate (melva@isp.nsc.ru).

Abstract: The problem of an increase of a parallelization potential of tasks in the computing system without an extension of a number of its processors is investigated. It is reached by such change of the initial topology when the relations of adjacency of processors are supplemented, but methods of their addressing and routing in the system remain the same. Reasons of the problem statement of the "edge" scal-

ing, which is consisting in the using of the redundant ports of the current configuration of the system for creation of links between the most remote processors is given in the article. For systems with the hypercubic topology the functions of a limit parallelization and of a topological scalability on tasks in which length of a path between branches is limited by volumes of the processed data and the used technology of a network, was obtained. The corresponding schedules are provided, and their analysis is given.

Keywords: hypercubic computing system, supercomputer, parallelism, topological scalability, edge scaling.

Статья представлена к публикации
членом редакционной коллегии ...заполняется редактором...

Поступила в редакцию ...заполняется редактором...
Опубликована ...заполняется редактором...