

Власов А.А.
Организация вычислительного процесса в ПВС с
гиперкубовой топологией

АННОТАЦИЯ. В докладе рассматриваются достоинства и недостатки основных видов топологий гиперкуба (двоичный гиперкуб(ДГ), обобщенный гиперкуб(ОГ), обобщенный кольцевой гиперкуб(ОКГ), обобщенный кольцевой гиперкуб дополненный кольцевыми магистралями(ОКГМ)). Проводиться сравнение их параметров. Дается оценка пропускной способности в зависимости от числа измерений и узлов в структуре. Показано, что ОКГМ характеризуется постоянным диаметром. Показано, что ОКГ, ОГ, ОКГМ по сравнению с ДГ дают возможность сравнительно просто реализовать волновой способ организации вычислений. В качестве критерия целесообразности и эффективности волнового способа организации вычислений выступают параметры модели задачи

Ключевые слова и фразы: топология, гиперкуб, модель задачи, волновой способ вычислений.

В настоящее время все большее применение в ПВС находят топологии типа гиперкуб. Топология двоичный гиперкуб(ДГ) нашла свою реализацию как в существующих ПВС, так и в проектах, которые предполагается реализовывать в «железе». Также под разными названиями реализованы различные варианты топологии обобщенной кольцевой гиперкуб(ОКГ). О реализации топологии обобщенный гиперкуб(ОБ) в настоящее время неизвестно.

Особенностью гиперкубовых структур является то, что они характеризуются высокой степенью регулярности, возможностью масштабирования и наращивания количества процессоров в системе. К недостаткам следует отнести увеличение количества связей при увеличении числа процессоров и достаточной сложности поддержания когерентности памяти.

Структуры ДГ, ОГ, ОКГ задаются следующими соотношениями полученными в работе[1]. Здесь же приводятся их параметры (Табл.1).

Двоичный гиперкуб (ДК) размерности $Co(n)$ ($N = 2^n$ число вершин): каждая вершина v задана двоичным числом $p(v) = (p_0 \dots p_i \dots p_{n-1})$ и ребрами соединены те и только те пары вершин v, u , для которых $p(v) \oplus p(u) = 1$, где $p(v) \oplus p(u)$ обозначает операцию покомпонентного суммирования по $\text{mod } 2$ n -мерных двоичных векторов; можно также определить $Co(n)$ с помощью n функций вида $C_i(p_0 \dots p_{i-1} p_{i+1} \dots p_{n-1}) = p_0 \dots p_{i-1} p_{i+1} \dots p_{n-1}$.

Обобщенный гиперкуб (ОГ) размерности $n C_1(m_1, m_2, \dots, m_n)$ ($N = m_1 \times m_2 \times \dots \times m_n$). Для всех $l = 1, \dots, n, m_l > 1$ каждая вершина v задана числом со смешанным основанием $r(v) = (x_1 \dots x_l \dots x_n)$, где $0 \leq x_l \leq m_l - 1$, и вершина с номером $(x_1 \dots x_{l-1} x_{l+1} \dots x_n)$ соединена

ребром с теми и только теми вершинами, номера которых удовлетворяют $(x_1 \dots x_{l-1}x_l x_{l+1} \dots x_n)$ для всех $1 \leq l \leq n$, где x_l пробегает все целые значения между 0 и m_l-1 , за исключением x_l . Таким образом, обобщенный гиперкуб — это n -мерный гиперкуб с m_l вершинами в каждом измерении.

Обобщенный кольцевой гиперкуб (ОКГ) размерности n $C1g$ (m_1, m_2, \dots, m_n) ($N = m_1 \times m_2 \times \dots \times m_n$). Вершины заданы так же как и в случае $C1$, но каждая вершина с номером $(x_1 \dots x_{l-1}x_l x_{l+1} \dots x_n)$ соединена ребром с вершинами $(x_1 \dots x_{l-1}(x_l+1) \bmod m_l x_{l+1} \dots x_n)$ и $(x_1 \dots x_{l-1}(x_l-1) \bmod m_l x_{l+1} \dots x_n)$ для всех $1 \leq l \leq n$.

Таблица 1. Основные параметры гиперкубовых топологий

Тип Структуры	Диаметр D	Степень s	Стоимость C=sD	Среднее расстояние $d = \frac{\sum_{r=1}^D r N_r}{N} - 1$	Средняя плотность Сообщения $p = \frac{dN}{Z}$	Общее число связей Z	Отказоустойчивость F
ДГ	$\log_2 N$	$\log_2 N$	$\log_2^2 N$	$\approx 0,5 \log_2 N$	~ 1	$0,5N \log_2 N$	$(\log_2 N) - 1$
ОГ	n	$1,5 \log_2 N$	$0,75 \log_2^2 N$	$0,375 \log_2 N$	$\sim 0,5$	$\frac{N}{2} \sum_{i=1}^n (m_i - 1)$	$(1,5 \log_2 N) - 1$
ОКГ	$\sum_{i=1}^n [m_i/2]$	2n	$0,889 \log_2^2 N$ При $n = [\log_8 N]$	$0,667 \log_2 N$ при $m_i = 8$ для всех i	2	$0,334N \log N$	$(0,667 \log_2 N) - 1$

ОКГ, как структура, представляется по сравнению с ДГ, более подходит для реализации на ней большого класса задач. В случае n -мерного ОКГ исходный информационный граф алгоритма может быть декомпозирован на планарные графы, размещаемые каждый в своей плоскости. Наличие связей в третьем измерении кольцевые связи позволяет организовать обмен информацией между ветвями при реализации параллелизма независимых ветвей. ОГ присущи практически все достоинства и недостатки структуры с топологией полного графа. Главным недостатком ОГ является быстрый рост числа связей с увеличением числа процессоров в ПВС.

В [2] предложена топология ОКГ дополненная магистралями (ОКГМ) по всем измерениям. Включение магистралей в структуру

устраняет многие недостатки рассмотренных ранее топологий [2]. В результате получается структура со смешанным типом связей [3], которая обеспечивает сравнительную простоту трансляции, маршрутизации сообщений и когерентности памяти узлов ПВС. В [4] предложена архитектура ПВС с масштабируемой структурой на основе ОКГМ. В момент оформления работ [3,4] реализация этой структуры представлялась проблематичной, вследствие технической сложности организации межпроцессорного интерфейса, в настоящее время, это вполне выполнимо.

На графике (рис. 1) представлены зависимости диаметров рассматриваемых структур от количества узлов.

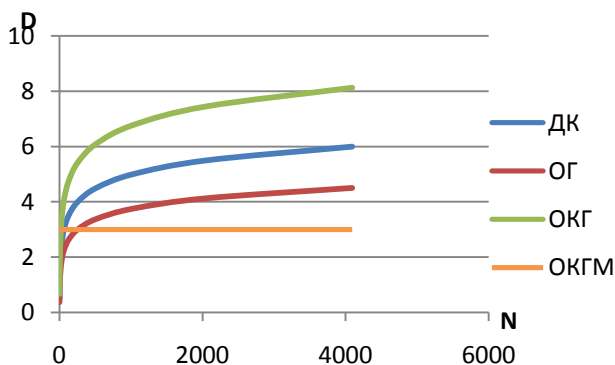


Рисунок 1. Зависимость диаметра структуры от числа узлов

На (рис.2) представлена зависимость потенциальной пропускной способности коммутационных структур от числа межпроцессорных связей.

В данном случае под потенциальной пропускной способностью понимается общее число межпроцессорных связей при условии дуплексного режима передачи данных.

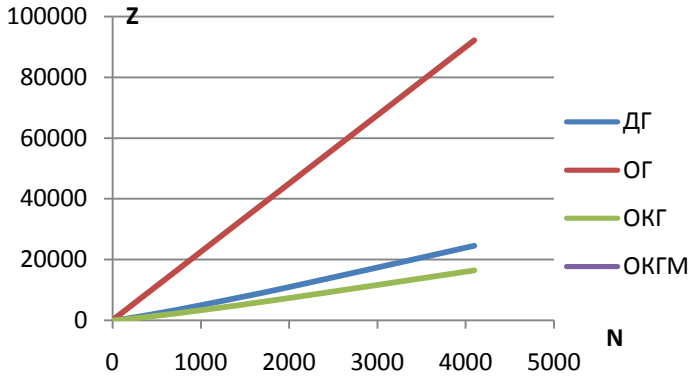


Рисунок 2. Зависимость количества связей от числа узлов

Показатель пропускной способности и скорость установления связи между обрабатывающими устройствами определяет возможность и целесообразность решения сложных задач на конкретной архитектуре вычислительных средств. Для сравнительной оценки эффективности рассматриваемой топологии ОКГМ был проведен анализ параметров рассматриваемых коммутационных структур, имеющих высокие показатели пропускной способности, возможности по масштабируемости и обладающих высокой степенью регулярности в зависимости от числа узлов на основе их исследования с помощью системы моделирования [5].

Сравнительные характеристики графов этих структур представлены на рис. 3-4.

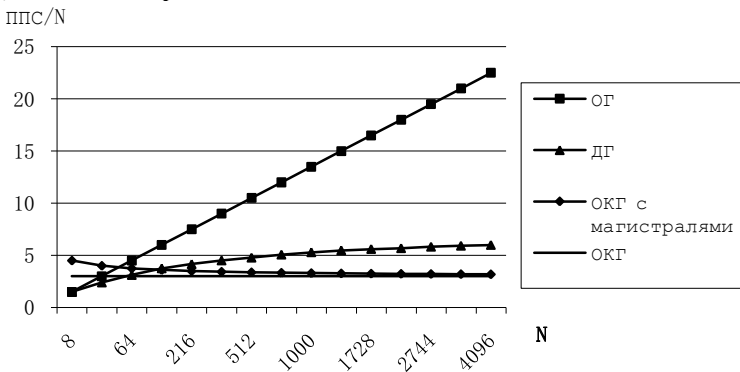


Рис. 3. Отношение ППС к числу узлов

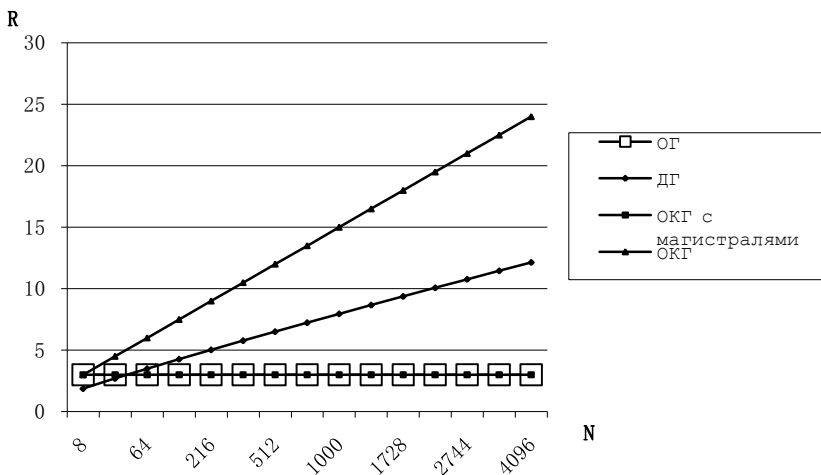


Рис. 4. Радиус КС

На рисунках применены следующие обозначения: ППС - потенциальная пропускная способность; R - радиус структуры.

ПВС на основе топологии типа гиперкуб обеспечивает возможность организации волновых вычислений. Некоторые задачи вычислительной математики непосредственно позволяют организовать этот способ вычислений [6]. В общем случае информационный граф алгоритма решения задачи может быть подвергнут расслоению и размещен на соответствующих плоскостях гиперкуба. С этой целью могут быть выделены фрагменты задачи имеющие сравнительно небольшой объем межфрагментных обменов. На основе этих фрагментов сформированы функциональные автономные блоки, а в ряде случаев макро и мегаоперации [7]. При построении коммуникационной среды ПВС на основе коммуникационного процессора предложенного в [8] возможна коммутация каналов и настройка межпроцессорных связей в соответствии с графами алгоритмов фрагментов. При этом может быть выполнена настройка на эти фрагменты в динамике при использовании магистральных связей. В результате такая ПВС приближается по своим свойствам к идеальной ЭВМ. Для оценки эффективности структур ВС при решении определенного класса задач необходимо такое описание параметров алгоритмов этих задач, которое отражало бы их свойства наиболее существенные при реализации на параллельных ВС, носило достаточно общий характер

и однозначно соответствовало элементам архитектуры. Такое описание может быть представлено в виде модели задачи [6].

Наиболее общей формой представления параллельных (и последовательных) алгоритмов является информационно-управляющий граф алгоритма (ИГ), который отражает зависимость по данным между операторами алгоритма и безусловные и условные переходы в программе. Такой граф в неявной форме содержит все виды параллелизма для выбранного метода решения задачи.

Модель задачи строится на основе анализа графа алгоритма программы и включает все основные формы параллелизма и состоит из участков трех типов: скалярных (СК), участков с параллелизмом независимых ветвей (ВТ), векторных участков (ВК) (естественный параллелизм). Каждая из ветвей при этом содержит параллелизм смежных операций.

Таким образом, модель задачи включает в себя доли вычислений, входящих в соответствующие различные формы параллелизма и ширину этих форм параллелизма, количество обменов при реализации этих форм параллелизма.

С учетом межярусных связей модель задачи представляется в виде десятки:

$MЗ = \langle \Delta СК, \Delta ВТ, \Delta ВК, \Delta' ВТ, \Delta' ВК, \alpha СК, \alpha ВТ, \alpha ВК, \alpha' ВТ, \alpha' ВК \rangle$,

где $\Delta СК, \Delta ВТ, \Delta ВК$ – доля скалярных операций, доля операций с параллелизмом независимых ветвей, доля операций векторных участков соответственно; $\Delta' ВТ, \Delta' ВК$ – доля операций, связанных с обменом, при реализации ветвей и векторов соответственно;

$\alpha ВТ, \alpha ВК$ – ширина параллелизма ветвей и векторных участков соответственно;

$\alpha' ВТ, \alpha' ВК$ – ширина пакета связей (число связей при обменах между ветвями или векторами соответственно).

Время выполнения задачи в соответствии с этой моделью в условных тактах:

$T = t(\Delta СК / \alpha СК + \Delta ВТ / (\alpha СК \alpha ВТ) + \Delta ВК / \alpha ВК) + t' ВТ \Delta' ВТ / \alpha' ВТ + t' ВК \Delta' ВК / \alpha ВК$,

где t – среднее время выполнения вычислительной операции; $t' ВТ, t' ВК$ – среднее время выполнения операции обмена для ветвей и векторов соответственно.

В качестве обобщенного критерия при реализации фрагмента может быть взят критерий времени выполнения алгоритма. Так как алгоритм реализуется в разных системах команд, то объем вычислений и обменов целесообразно представить в стандартных операциях, а обмен, например, в байтах.

Поскольку быстродействие обрабатывающих устройств и скорость передачи данных зависит от технологии СБИС, то для

оценки времени, необходимого для реализации алгоритма на ВС, представим время выполнения операции и передачи элемента данных в условных тактах. При этом сделаем следующее допущение, что время передачи одного элемента данных равно одному такту, тогда минимальное время реализации алгоритма соответствующее критическому пути

$$T \leq T_0 + T_{\Pi},$$

где T_0 - время необходимое для преобразования данных критического пути в ЯПФ алгоритма; T_{Π} - время, необходимое для передачи данных критического пути от ввода исходных данных до вывода результатов.

$$T_0 = \sum_{i=1}^n \varepsilon_i \sum_{k=1}^K t_k a_{i,b,k}$$

$$T_{\Pi} = \sum_{i=1}^n \varepsilon_i \sum_{k=1}^K t_{\Pi} v_{i_s, b_s}$$

где $t_k = \beta_{k\tau}$ - время выполнения k - команды; $t_n = \gamma\tau$ - время передачи одного слова данных; β_k - число тактов, необходимое для выполнения k -ой команды; γ - количество тактов необходимых для передачи слова данных; τ - длительность такта.

$$\varepsilon_i = \begin{cases} 1, & \text{если } i\text{-я вершина ЯПФ входит в критически й путь} \\ 0, & \text{если } i\text{-я вершина ЯПФ не входит в критически й путь} \end{cases}$$

Нестрогое равенство в этих выражениях указывает на то, что в принципе возможно, в рамках вычислений и при наличии соответствующих аппаратных средств, совмещение этапов обработки и передачи данных.

С учетом изложенного времени реализация алгоритма на реальной ВС будет определяться формулой сходной с выражением.

$$T = \sum_{i=1}^n \varepsilon_i \sum_{k=1}^K t_k a_{i,b,k} + V_0 \cdot t_{\Pi}$$

Данная формула в значительной степени учитывает систему команд поскольку длина программы и количество промежуточных данных определяется системой команд.

При организации вычислительного процесса, в виде потока фрагментов, включение коммуникационного процессора в состав узла ПВС обеспечивает возможность настройки на конкретные графы их алгоритмов.

Библиографический список

1. L. N. Bhuyan and D. P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," IEEE Trans. Comput. C-33, 323–333 (1984).

2. Власов А.А. Коммуникационная среда на основе однородной коммутационной структуры / А.А. Власов, П.В. Михеев // Труды междунар. конф. по информ. сетям и системам. ICINAS-2000.– СПб., 2000. С. 439-452.

3. Артамонов Г.Т., Тюрин В.Д. Топология сетей ЭВМ и многопроцессорных систем. – М.: Радио и связь, 1991. – 248 с.

4. Власов А.А. Параллельная вычислительная система с масштабируемой структурой // Тр. науч. конф. по итогам н.-и. работ Мар. гос. техн. ун-та. Йошкар-Ола, 19-23 марта, 2001. / Йошкар-Ола: МарГТУ, 2001. - С. 54-70.- - Деп. в ВИНТИ 11.02.2002 № 277-В2002.

5. Система моделирования коммуникационной среды типа гиперкуб. Свидетельство об официальной регистрации программы для ЭВМ №2003611049 Зарегистрирована в реестре программ для ЭВМ. Москва 05.05.2003

6. Гергель В.П., Стронгин, Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем.// Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2003. 184 с.

7. Власов А.А. Формирование макрокоманд на основе выделения функциональных автономных блоков. //Тр. науч. конф. по итогам н.-и. работ Мар. гос. техн. ун-та. Йошкар-Ола, 24-28 апр., 2000. Йошкар-Ола: МарГТУ, 2000. - С. 78-92. - Деп. в ВИНТИ 2001.

8. Власов А.А., Михеев П.В. Коммуникационный процессор, патент RU 2260841 С2, 20.09.2005