

На правах рукописи

ЭЛЬ ИДРИССИ ЛАМГХАРИ МУСТАФА
(МАРОККО)

УДК 681.324

**СПОСОБЫ И СРЕДСТВА ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ
ГИПЕРКУБИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

- Специальность 05.13.08 - "Вычислительные машины, системы и сети,
элементы и устройства вычислительной
техники и систем управления".
05.13.09 - "Математическое и программное обеспечение
вычислительных машин, комплексов и сетей".

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени
кандидата технических наук

Киев - 1996 г.

АВ 34.930

Диссертация является рукописью.
Работа выполнена в Национальном техническом университете Украины
"Киевский политехнический институт" на кафедре
вычислительной техники

ЛНБ України ім. В. Стефаника
00754660 (S)

- Научный руководитель: доктор технических наук,
профессор,
Луцкий Георгий Михайлович.
- Консультант: кандидат технических наук,
доцент,
Корочкин Александр Владимирович.
- Официальные оппоненты: доктор технических наук,
профессор,
Молчанов Александр Артемьевич,

кандидат технических наук,
доцент,
Жуков Игорь Анатольевич.

Ведущая организация: Институт проблем моделирования
в энергетике НАН Украины.

Защита состоится 17 июня 1996 г. в 14³⁰ часов на заседании
специализированного Совета Д 01.02.06 в Национальном техническом
университете Украины "Киевский политехнический
институт" (г. Киев, пр. Победы, 37, корп. 18, ауд.306)

Отзывы на автореферат в двух экземплярах, заверенные печатью уч-
реждения, просим направлять по адресу:
252056, г. Киев, пр. Победы, 37, Ученому секретарю КПИ.

С диссертацией можно ознакомиться в библиотеке Киевского политехни-
ческого института.

Автореферат разослан 17 мая 1996 г.

Ученый секретарь
специализированного совета,
доктор технических наук

профессор

Бузовский О.В. Бузовский

ЛНБ ім. В. Стефаника
АН України

АВ - 34.950

АННОТАЦИЯ

Целью диссертационной работы является повышение эффективности гиперкубических вычислительных систем на основе развития структурной организации и совершенствования методов программирования.

Для достижения указанной цели в диссертации решаются следующие задачи:

1. Топологический анализ структур иерархических гиперкубических систем; поиск оптимальных топологий таких систем.
2. Анализ особенностей программного обеспечения вычислительных систем с гиперкубической топологией; поиск путей совершенствования процесса разработки программ для систем с архитектурой гиперкуба.
3. Разработка и исследование методики программирования для развитых гиперкубических систем на основе абстракций и спецификаций; исследование возможностей ее реализации с помощью современных языков программирования.
4. Оценка эффективности предлагаемых подходов при разработке пакета программ для развитых гиперкубических систем с различной структурной организацией.

Автор защищает следующие положения и результаты:

1. Способы и средства поиска оптимальных топологий иерархических гиперкубических систем с заданными топологическими характеристиками.
2. Методику разработки программного обеспечения для иерархических гиперкубических систем на основе абстракций, учитывающих особенности вычислительных процессов в гиперкубе и направленных на оптимизацию межузлового взаимодействия.
3. Результаты экспериментальной проверки предложенной методики, выполненной при разработке программного обеспечения для гиперкубических систем.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В настоящее время значительное внимание уделяется поиску путей и средств повышения эффективности структурной организации параллельных вычислительных систем, а также программирования в таких системах.

Структурная организация, использующая топологию гиперкуба, широко применяется при проектировании вычислительных систем и является основой для построения высокопроизводительных систем, обладающих рядом привлекательных свойств (однородность, симметричность, оптимальный баланс между количеством связей и их стоимостью, логарифмическая зависимость диаметра системы от количества процессоров). Недостатками гиперкубических топологий фиксированное для каждого k - гиперкуба число процессоров (2^k), жесткость структурной организации, избыточность связей между процессорами, зависимость топологических характеристик от размерности гиперкуба. Это предопределяет снижение эффективности гиперкубических систем при решении ряда задач. Поэтому актуальной является проблема развития структурной организации гиперкубических систем, сохраняющих их достоинства и устраняющих указанные недостатки. В диссертационной работе подход к решению поставленной проблемы связывается с иерархическими гиперкубическими топологиями.

Наряду с задачей совершенствования структурной организации гиперкубических систем все большее значение приобретает проблема разработки программ для них. Опыт использования систем с архитектурой типа гиперкуб показал, что их эффективность во многом зависит от правильно организованного программирования, учитывающего особенности структуры системы. Проектированию программ для вычислительных систем в последнее время уделяется большое внимание, что привело к появлению ряда методик программирования (Yordan, Mascot, Jackson Structured Design). Однако эти методики имеют узкую направленность, предполагают специфичные программные средства и не всегда позволяют учитывать модель вычислений, принятую в вычислительной системе. Поэтому в диссертационной работе рассматриваются вопросы поиска способов и средств проектирования программ, направленных на повышение эффективности разработки программного обеспечения для иерархических гиперкубических систем.

Методы исследования. Для решения поставленных задач использовались методы теории графов, комбинаторики, теории проектирования вычислительных систем, теории организации вычислительных процессов, теории параллельного программирования, теории численных методов решения задач линейной алгебры.

Научная новизна. Предложен вариант решения задачи выбора иерархических гиперкубических структур с заданными топологическими характеристиками. Определены требования к методике разработки программного обеспечения, учитывающей особенности структурной организации таких систем. Предложены и обоснованы способы и средства разработки программ для развитых гиперкубических систем на основе абстракций различного вида, позволяющие автоматизировать программирование в масштабируемых иерархических гиперкубических вычислительных системах.

Практическая ценность диссертационной работы заключается в том, что полученные в ней результаты позволяют обоснованно решать важные практические задачи выбора структурной организации иерархических гиперкубических систем, учитывающей особенности решаемых задач, и повышения эффективности программирования для вычислительных систем с развитой гиперкубической структурой.

Достоверность теоретических результатов подтверждается экспериментальной проверкой возможности поиска структур иерархических гиперкубических систем с заданными топологическими характеристиками и применением предложенной методики для разработки системного и прикладного программного обеспечения.

Реализация работы. Основные результаты диссертационной работы использованы в учебном процессе на кафедре вычислительной техники в курсах "Вычислительные системы" и "Параллельное программирование".

Публикации. По теме диссертации опубликовано 3 работы.

Структура и объем работы. Диссертационная работа состоит из введения, пяти глав, заключения и приложения, изложенных на 110 страницах машинописного текста, содержит 3 таблицы, 63 рисунка, список литературы из 60 наименований и приложения на 18 страницах.

Во введении обосновывается актуальность темы диссертационной работы, формулируется цель и задачи исследований, основные положения, выносимые на защиту.

В первой главе на основе анализа гиперкубических структур и программного обеспечения определяются пути повышения эффективности структурной организации и методов разработки программ в гиперкубических системах, осуществляется обоснование выбора и уточненная постановка задач исследований, проводимых в диссертации.

Во второй главе выполнен сравнительный анализ иерархических гиперкубических структур, получены аналитические зависимости параметров, рассмотрена задача поиска оптимальных структурных организаций.

В третьей главе анализируются и исследуются возможности использования абстракций различного вида для повышения эффективности разработки программ в гиперкубических вычислительных системах с развитой структурной организацией.

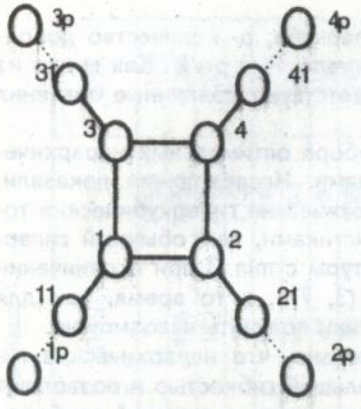
В четвертой главе рассмотрена реализация предложенных абстракций с помощью средств языка программирования Ада.

В пятой главе анализируется эффективность предложенной методики проектирования программного обеспечения при разработке прикладного пакета программ для иерархических систем с различной структурной организацией.

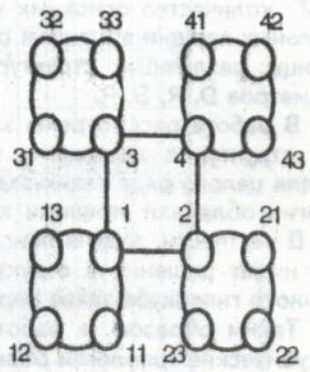
СОДЕРЖАНИЕ РАБОТЫ

Недостатки, присущие гиперкубическим структурам, в значительной мере можно преодолеть в иерархических гиперкубических структурах, в которых с каждым узлом гиперкуба отождествляется определенная структурная организация (Рис. 1).

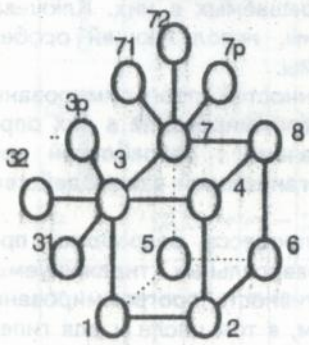
С целью исследования возможностей иерархических гиперкубических структур в работе проведен анализ структур, в которых с узлами отождествляются типовые топологии: линейная, кольцевая, звездообразная, кубическая. Получены аналитические зависимости топологических характеристик этих структур (D, R, S, P), выполнен их сравнительный анализ, рассмотрены задачи поиска структур, удовлетворяющих заданным требованиям. В таблице 1 представлены возможные структуры иерархических гиперкубов $HX(l,p)$ для заданного числа узлов $P=64$,



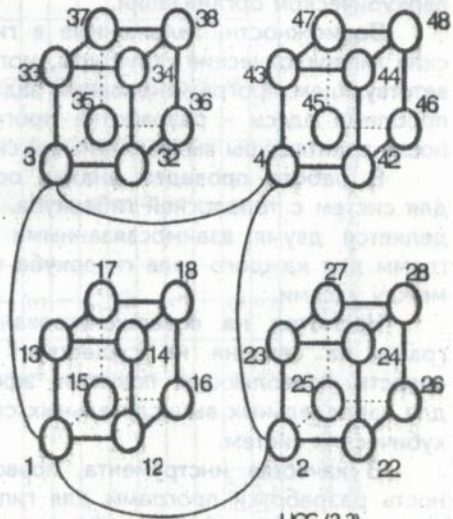
HL(2,P)



HR(2,3)



HS(3,P)



HOC(2,3)

Рис. 1

2-6-1884

где 2^1 - количество основных узлов гиперкуба, p - количество дополнительных вершин в каждом основном узле, $P = p \cdot 2^1$. Как видно из таблицы, различным структурам соответствуют различные значения параметров D, R, S, P .

В работе рассмотрены задачи выбора оптимальных иерархических структур с заданными параметрами. Исследования показали, что для целого ряда таких задач иерархические гиперкубические топологии обладают лучшими характеристиками, чем обычный гиперкуб. В частности, задача поиска структуры с $\min D$ при ограничении $S=4$ имеет решение в структуре HL (3, 7), в то время, как для обычного гиперкуба такие характеристики получить невозможно.

Таким образом, в работе обосновано, что иерархические гиперкубические топологии обладают большей гибкостью и позволяют осуществить выбор соответствующей структуры, в которой наиболее полно учитываются особенности решаемых задач и информационных связей в них, сохраняя в той или иной мере характеристики гиперкубической организации.

Возможности, заложенные в гиперкубические и в иерархические гиперкубические структуры, могут быть реализованы при соответствующем программировании задач, решаемых в них. Ключевая проблема здесь - разработка программы, использующей особенности архитектуры вычислительной системы.

В работе проведен анализ особенностей программирования для систем с топологией гиперкуба. Программирование в них определяется двумя взаимосвязанными задачами: разработкой программ для каждого узла гиперкуба и организацией взаимодействия между узлами.

Несмотря на совершенствование процесса разработки программ на сегодня не существует универсальных тиражируемых средств, позволяющих повысить эффективность программирования для параллельных вычислительных систем, в том числе и для гиперкубических систем.

В качестве инструмента, позволяющего повысить эффективность разработки программ для гиперкубических систем, в работе рассматривались абстракции различного вида. Существующие виды абстракций широко используются в языках программирования и являются мощными средствами создания эффективных программ. Задачей этапа проектирования программы является выполнение декомпозиции исходной задачи на ряд модулей.

Таблица 1.

N	Вид структуры в узлах	I	p	D диаметр	S степень	R число ребер	D*S	D*R	S*R
1	Стандартная (H)	6	-	6	6	192	36	1152	1152
2	Линейная (HL)	1	31	63	2	63	126	3959	126
3		2	15	34	3	64	102	2176	192
4		3	4	17	4	68	68	1156	272
5		4	3	10	5	80	50	800	400
6		5	1	7	6	112	42	784	672
7	Кольцевая (HR)	1	31	49	3	61	47	2989	183
8		2	15	18	4	60	72	1080	240
9		3	7	11	5	60	55	660	300
10		4	3	8	6	64	48	512	384
11		5	1	6	7	80	42	480	560
12	Звезда (HS)	1	31	3	32	63	96	192	2048
13		2	15	4	17	64	68	272	1156
14		3	7	5	10	68	50	400	700
15		4	3	6	7	80	42	560	784
16		5	1	7	6	192	42	784	672
17	Квадратная (HC)	4	2	8	6	96	48	768	576
18	Кубическая (HCC)	3	3	9	6	108	54	972	648

I - размерность I - куба

p - размерность встраиваемой вместо вершины топологической сети

В ходе декомпозиции необходимо учесть какие модули наиболее подходящие и какие способы увеличивают вероятность того, что их объединение даст возможность решить поставленную задачу. Абстракция является эффективным способом проведения декомпозиции, предполагая целенаправленный выбор компонентов при декомпозиции.

Различают абстракцию через параметризацию и абстракцию через спецификацию. Абстракция на основе параметризации позволяет с помощью механизм параметров описывать неограниченный набор вычислений одним модулем и является средством повышения универсальности программ. Абстракция на основе спецификации позволяет абстрагироваться от процесса вычислений, описанных в теле модуля. Это достигается путем задания для каждой абстракции спецификации, описывающей ее работу. Несущественность способа реализации в абстракции через спецификацию позволяет переходить к новой реализации без внесения изменений в программу. Абстракция на основе спецификации наделяет программу двумя отличительными особенностями: локальностью, при которой реализация каждой абстракции выполняется без анализа реализаций других абстракций, и модифицируемостью, которая позволяет упростить модификацию программы, так как изменения реализации абстракции без изменения спецификации не влияют на всю программу.

В работе рассмотрены возможные виды абстракций и выделена абстракция процесса, как основополагающая при проектировании программ в гиперкубе. На основе анализа вычислительных процессов в гиперкубе были определены требования к абстракции процесса, учитывающие их особенности. На Рис. 2

PROCESS имя

RECEIVE -- входные данные
 SEND -- выходные данные
 PRIOR -- приоритет
 EFFECT -- выполняемые функции

·
·
·

Рис. 2

представлена спецификация абстракции процесса, включающая предложения, связанные, в первую очередь, с организацией эффективного взаимодействия процессов, характерного для гиперкуба.

Предложение RECEIVE описывает информацию о типе и количестве данных, которые принимаются данным процессом от других процессов (входные данные).

Предложение SEND задает информацию о типе и количестве данных, передаваемых из одного процесса в другие процессы (выходные данные).

Предложение PRIOR описывает приоритет процесса, который будет учитываться системой управления процессами.

Предложение EFFECT описывает работу процесса со значениями, охваченными предложениями SEND и RECEIVE. Предлагаемая абстракция процесса использует абстракции через параметризацию и через спецификацию. Параметризованная абстракция процесса представляет собой класс взаимосвязанных абстракций, определяемых с помощью одной спецификации, и позволяет достичь обобщения и расширения возможностей абстракции процесса.

Разработка программ разделяется на несколько этапов: анализ требований, проектирование, реализация, тестирование, эксплуатация, модификация, сопровождение. Назначение этапа анализа требований заключается в создании спецификации требований, которая будет являться исходным пунктом для этапа проектирования. На этапе проектирования производится модульная декомпозиция, удовлетворяющая спецификации требований. Ее результат - структура программы, состоящей из набора модулей, удовлетворяющая ряду требований.

Методика проектирования, основывающаяся на абстракции, предполагает выделение на этапе анализа требований одной или нескольких целевых абстракций. Для каждой из них затем выполняются следующие действия: определение вспомогательных абстракций для реализации целевой и проведения декомпозиции; определение свойств каждой вспомогательной абстракции; определение реализации целевой абстракции через вспомогательные.

Зависимость модулей представляется структурной схемой, на которой фиксируются абстракции и связи между ними (Рис.3). Структурная схема зависимости модулей является направленным графом; вершины графа представляют абстракции, ребра показывают, какие абстракции используются при реализации новых абстракций. Тип вершины зависит от вида используемой абстракции (данных, процедур, процесса). Целевая абстракция использует или зависит от вспомогательных абстракций. Процедура используется через обращение к ней, тип данных - через вызов его операций. Граф может включать циклы, если имеет место рекурсия. Для модулей типа процесс (модули А, В, F, G, Е, К) связи означа-

ют взаимодействия, во время которого осуществляется передача (прием) сообщений, что дополнительно указывается на структурной схеме. Дальнейшая детализация схемы может быть направлена на указание подробностей взаимодействия процессов: фиксированный или условный порядок взаимодействия, а также зависящий от времени.

Анализ взаимодействия процессов может прогнозировать возникновение тупиковых ситуаций (deadlock), причиной которых может стать появление на структурном графе прямого или косвенного цикла во взаимодействии модулей типа процесс.

Использование абстракций улучшает этап проектирования; поиск и выбор абстракций соответствующего вида позволяет учесть особенности решаемой задачи, абстрагирование от подробностей - сосредоточить внимание на связях модулей и особенно на взаимодействии процессов, так как это во многом будет определять как работоспособность, так и эффективность программы в гиперкубической системе.

Подробная структурная схема модульной зависимости абстракций является результатом выполнения этапа проектирования и основой для последующего этапа реализации, выполняемого с помощью языка программирования, средствами которого реализуются абстракции, использованные при проектировании.

Для реализации рассмотренных абстракций различного вида в работе использовался язык программирования Ада. Ада - это язык высокого уровня универсального назначения, созданный в 1984 году по заказу Министерства обороны США. В 1995 году появился новый стандарт языка Ада - 95, который прошел полную международную сертификацию (ISO/ IEC 3652 : 1995 (95-02-15) ANSI (95-04-10), FIPS 119-1 (95-07-10)). Язык основывается на модульном принципе построения программ, поддерживает оба направления их разработки (нисходящее и восходящее) и предназначен для построения больших программных систем.

Язык содержит средства высокого уровня для программирования параллельных алгоритмов. При этом используется модель параллельности Хоара (Hoare), в основе которой - взаимодействующие последовательные процессы, а для их синхронизации и обмена данными используются операторы, аналогичные операторам ввода - вывода.

В работе исследовались возможности языка Ада для программирования в развитых гиперкубических системах и реализации абстракций различного вида. В языке имеется развитый механизм подпрограмм в виде процедур и функций, удобный интерфейс для них в виде спецификаций. Для реализации процедурной абстракции существует явное разделение формальных параметров на виды (in, out, inout). Класс процедур через параметризацию может быть реализован использованием в качестве параметров типов и функций.

Пакеты - еще один вид программных модулей, из которых составляются программы. Пакет объединяет объекты и операции над ними и представляет собой механизм сокрытия информации и инкапсуляции данных. Спецификация пакета определяет его возможности, тело пакета реализует эти возможности. Механизм пакетов позволяет реализовать абстракции данных, однако не в полной мере, так как только частично поддерживает определение абстрактных типов данных.

При реализации абстракций данных определенный интерес представляют приватные (private) и ограниченные приватные (limited private), позволяющие скрыть детали реализации типов, объявленных в спецификации пакета. Скрытая реализация вынуждает оперировать над объектами личного типа, используя только операции, предоставленные вместе с личным типом и операциями, реализующими этот тип.

Различные виды абстракции на основе параметризации в Аде реализуются через механизм настраиваемых модулей, позволяющий повысить универсальность модулей.

Эффективность абстракций различного вида зависит от надежности обеспечения ее работы, соответствующей спецификации. Это соответствие обеспечивается различными методами, в том числе использованием механизма исключительных ситуаций. В Аде имеется универсальный механизм исключений, включающий описание и обработку исключений. Его использование локально внутри абстрактных модулей при их реализации гарантирует, что выполнение модуля будет осуществляться в соответствии со спецификацией.

Абстракция процесса в Аде реализуется через специальный вид модулей - задачи. Задача имеет собственную "нить" управления и способна выполняться параллельно с другими задачами внутри одной программы. Подобно другим видам программных модулей, задача состоит из спецификации и тела. Спецификация задачи определяет интерфейс для взаимодействия с другими задачами, которое выполняется на основе развитого механизма randevu. Средства, допускаемые в спецификации задачи, позволяют в значительной мере описать предложения шаблона абстракции процесса (Рис. 2) при ее реализации языком Ада. Предложения SEND и RECEIVE охватываются операторами ENTRY, в которых описывается тип, объем и направление передаваемой (принимаемой) информации между процессами; предложение PRIOR реализуется через указание компилятору (pragma PRIORITY), с помощью которого устанавливается и может быть изменен динамически (Ада - 95) приоритет процесса; предложение EFFECT реализуется в теле задачи как традиционными средствами, так и средствами, предназначенными для организации взаимодействия процессов (accept, select).

Параметризация процессов может выполняться через абстракцию данных и реализовываться через механизм настраиваемых пакетов; класс идентичных процессов строится на основе задачного типа (task type).

Отличительной особенностью всех видов модулей в Аде, из которых возможно построение программы, является наличие удобного интерфейса к ним в виде соответствующих спецификаций, задаваемых независимо от тел модулей. Структура программы, состоящей из модулей, может быть задана и описана через их спецификации, что является важным при проектировании на основе использования абстракций различного вида.

Другая важная особенность языка Ада, предоставляющая определенные удобства в плане реализации абстракций, это - раздельная компиляция. Раздельная компиляция является эффективным средством поддержки нисходящей разработки программ, лежащим в основе методов проектирования на основе абстракции, а также последующей модификации программы.

Для реализации абстракций, представленных на Рис. 3, используются: пакеты для модулей С, D, М, Н, процедуры - для модулей D, Н, М, задачи - для модулей А, В, F, E, G, К. На Рис. 4 представлена подробная схема взаимодействия модулей при реализации с помощью языка Ада.

Таким образом, средства языка Ада создают основу для эффективной реализации абстракций различного вида при разработке программ для развитых гиперкубических систем. В работе предложенная методика разработки программ была исследована при разработке программного обеспечения для иерархических гиперкубических вычислительных систем с различной структурной организацией с использованием языка Ада. Разработка программ на основе абстракции позволила повысить эффективность одного из наиболее сложных этапов разработки параллельных программ - этапа модульной декомпозиции. Для каждого модуля подбиралась абстракция, учитывающая особенности исходной задачи. При реализации абстракций средствами языка Ада были решены задачи, важные для программ в гиперкубических системах:

- расширение механизма разделяемых переменных до механизма типа "критическая область" для узлов с общей памятью;
- предложен механизм конфигурирования в масштабируемых системах;
- разработан механизм переменных протоколов, позволяющий минимизировать количество передаваемой информации между процессами.

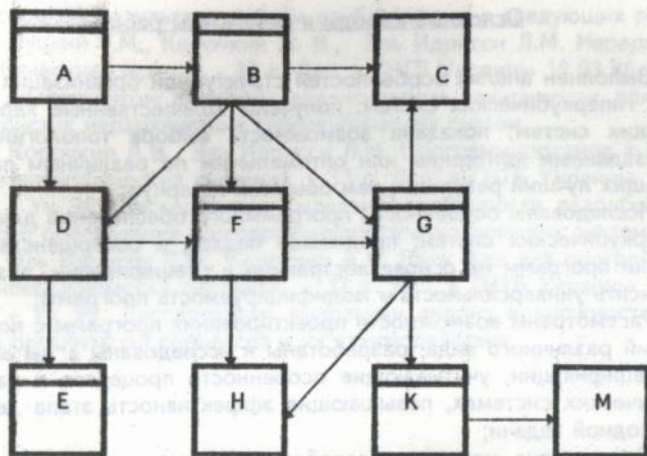


Рис. 3

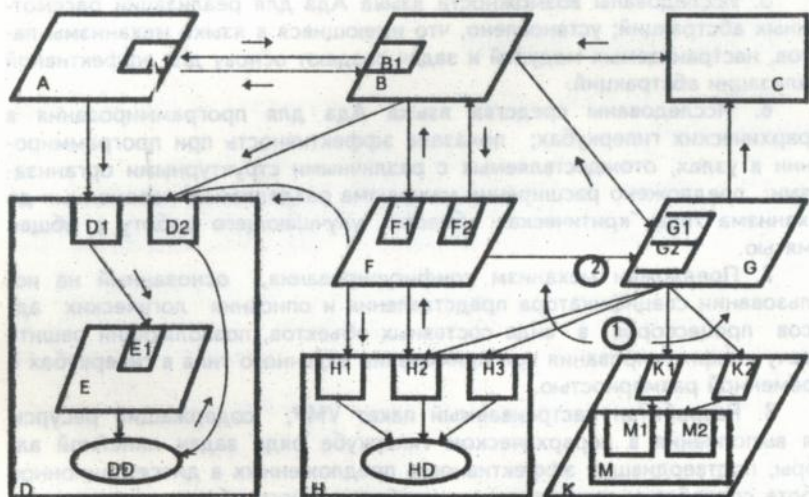


Рис. 4

Основные выводы и результаты работы.

1. Выполнен анализ особенностей структурной организации иерархических гиперкубических систем; получены количественные характеристики таких систем; показана возможность выбора топологий, отвечающих заданным критериям или оптимальным по различным параметрам, дающих лучший результат, чем обычный гиперкуб;

2. Исследованы особенности программного обеспечения для развитых гиперкубических систем; предложен подход к совершенствованию разработки программ на основе абстракции и спецификации, позволяющий повысить универсальность и модифицируемость программ;

3. Рассмотрены возможности проектирования программ с помощью абстракций различного вида; разработаны и исследованы виды абстракции и спецификации, учитывающие особенности процессов в развитых гиперкубических системах, повышающие эффективность этапа декомпозиции исходной задачи;

4. Предложена методика разработки программ для развитых гиперкубических систем, основанная на абстракции данных и процесса, позволяющая автоматизировать процесс проектирования программного обеспечения;

5. Исследованы возможности языка Ада для реализации рассмотренных абстракций; установлено, что имеющиеся в языке механизмы пакетов, настраиваемых модулей и задач создают основу для эффективной реализации абстракций.

6. Исследованы средства языка Ада для программирования в иерархических гиперкубах; показана эффективность при программировании в узлах, отождествляемых с различными структурными организациями; предложено расширение механизма разделяемых переменных до механизма типа "критическая область", улучшающего работу с общей памятью.

7. Предложен механизм конфигурирования, основанный на использовании спецификатора представления и описания логических адресов процессоров в виде составных объектов, позволяющий решить задачу конфигурирования при применении задачного типа в гиперкубах с переменной размерностью.

8. Разработан настраиваемый пакет VMP, содержащий ресурсы для выполнения в иерархическом гиперкубе ряда задач линейной алгебры, подтвердивший эффективность предложенных в диссертационной работе способов и средств создания программного обеспечения для вычислительных систем с развитой гиперкубической организацией.

Основные результаты работы опубликованы в следующих работах:

1. Луцкий Г.М., Корочкин А. В., Эль Идрисси Л.М. Иерархические гиперкубические системы. - 37 с. Деп. в ГНТБ Украины 19.03.96, № 757 - Ук 96. Диссертантом получены аналитические зависимости для оценки характеристик иерархических гиперкубических систем.

2. Корочкин А.В., Эль Идрисси Л.М. Программирование в иерархических гиперкубических системах. - 33 с. Деп. в ГНТБ Украины 19.03.96, № 756 - Ук 96. Диссертантом выделены особенности разработки программного обеспечения в иерархических гиперкубических системах.

3. Эль Идрисси Л.М., Корочкин А.В. Язык Ада для программирования в гиперкубических системах. - 37 с. Деп. в ГНТБ Украины 19.03.96, № 758 - Ук 96. Диссертантом выполнен анализ возможностей языка ада для организации процессов и их взаимодействия.

ЛНБ ім. В. Стефаника
АН України

Эль Идрисси Ламгхари Мустафа

Способы и средства повышения эффективности гиперкубических вычислительных систем

Работой является рукопись на соискание ученой степени кандидата технических наук по специальностям :

05.13.08 - "Вычислительные машины, системы и сети, элементы и устройства вычислительной техники и систем управления".

0513.09 - " Математическое и программное обеспечение вычислительных машин, комплексов и сетей".

Национальный технический университет Украины "Киевский политехнический институт"

Киев, 1996 г.

Целью диссертационной работы является повышение эффективности гиперкубических вычислительных систем на основе развития структурной организации и совершенствования методов программирования.

El Idrissi Lamghari Mustapha

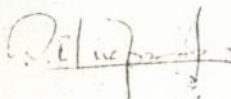
Methods and means of efficiency improvement of the hypercube computing system.

This scientific work is a manuscript to submit one's thesis for candidate's science in speciality 05.13.08 - Computers, systems and networks, elements and unites of computer technique and control systems; 05.13.09 - Mathematical and software of computers, complexes and networks.

The aim of thesis is to work and reserch methods and means of efficiency improvement of the hypercube computing system on the basis of development of structural organization and perfection methods of programming.

National Technical University of Ukraine "Kiev Politechnical Institute"
Kiev, 1996

Ключові слова: паралельні обчислювальні системи, гиперкубічні структури, програмне забезпечення, абстракція, декомпозиція, процес, мова Ада.



LIBRARY
IN THE
V. G. GORDEYEV
INSTITUTE

Подл. к печ. № 96. Формат 60×84¹/₁₆.
Бумага тип. № 1. Способ печати офсетный. Услови. печ. л. 998.
Услови. кр.-отт. 1,04. Уч.-изд. л. 1,0.
Тираж 100. Зак. № 6-1884.

Фирма «ВИПОЛ»
252151, г. Киев, ул. Воынская, 60.

УЧОНА

AB. 34.930

AB 34.930