82

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РСФСР

ЛЕНИНГРАДСКИЙ ОРДЕНА ЛЕНИНА ЭЛЕКТРОТЕХНИЧЕСКИЙ
ИНСТИТУТ ИМЕНИ В.И.УЛЬЯНОВА /ЛЕНИНА/

Для служебного пользо-
вания. Экз. № 90

Ю.Г. КОСАРЕВ

СИНТЕЗ ФУНКЦИОНАЛЬНЫХ СТРУКТУР ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С УПРАВЛЯЕМЫМИ СВЯЗЯМИ

Автореферат

диссертации на соискание ученой степени
доктора технических наук

С П Е Ц И А Л Ь Н О С Т Ь - 05.252 - Вычи-
слительная техника

Ленинград

1971

Работа выполнена в Институте математики Сибирского отделения АН СССР.

Официальные оппоненты:

1. Доктор технических наук, профессор А.В. Калиев.
2. Доктор технических наук С.Д. Пашкеев.
3. Доктор технических наук, профессор В.Б. Смолов.

Ведущее предприятие :

Институт кибернетики АН УССР, г. Киев.

Автореферат разослан "31" марта 1971 г.

Защита состоится "___" ________ 1971 г.

на заседании Совета по присуждению ученых степеней факультета автоматики и вычислительной техники Ленинградского ордена Ленина электротехнического института имени В.И. Ульянова /Ленина/, г. Ленинград, П-22, ул. Проф. Попова, 5 .

С диссертацией можно ознакомиться в библиотеке института.

Ученый секретарь Совета - Е.А. Чернявский

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РСФСР
ЛЕНИНГРАДСКИЙ ОРДЕНА ЛЕНИНА ЭЛЕКТРОТЕХНИЧЕСКИЙ
ИНСТИТУТ ИМЕНИ В.И.УЛЬЯНОВА /ЛЕНИНА/

Для служебного пользования. Экз. № 90

Ю.Г. КОСАРЕВ

СИНТЕЗ ФУНКЦИОНАЛЬНЫХ СТРУКТУР ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С УПРАВЛЯЕМЫМИ СВЯЗЯМИ

Автореферат

диссертации на соискание ученой степени
доктора технических наук

С П Е Ц И А Л Ь Н О С Т Ь - 05.252 - Вычислительная техника

Ленинград
1971

ВВЕДЕНИЕ

В ходе научно-технического прогресса непрерывно растет не только общий объем вычислений, но и вычислительная сложность задач. Отсюда вытекает необходимость постоянного повышения как суммарной, так и рекордной производительности вычислительной техники. Уровень рекордной производительности определяет время, качество, а нередко и самую возможность решения многих жизненно важных народно-хозяйственных, военных, политических, идеологических и других задач.

В настоящее время возникает все большее число проблем, для решения которых необходимы средства переработки больших объемов информации с производительностью 10^9 и выше операций в секунду.

Сложность разработки таких средств обуславливается двумя основными обстоятельствами:

– Известные методы повышения производительности, основывающиеся на увеличении быстродействия элементов, здесь оказываются малопригодными, так как предъявляют к параметрам элементов требования, намного превышающие возможности современной физико-технологической базы.

– Существующие методы построения математического обеспечения также оказываются недостаточными, в первую очередь потому, что с ростом производительности приходится резко увеличивать объем математического обеспечения, который у некоторых ЭВМ и без того достигает нескольких миллионов слов.

Для создания указанных средств вычислений с производительностью порядка миллиардов операций в секунду необходимы принципиально новые подходы как к логическому конструированию, так и к разработке математического обеспечения.

Около 10 лет назад автором совместно с Э.В. Евреиновым были выдвинуты и обоснованы принципы построения высокопроизводительных средств вычислений для решения задач с большим объемом перерабатываемой информации. Главные из этих принципов следующие: принцип параллельности, принцип однородности, принцип программной изменяемости структуры.

Указанные принципы легли в основу нового направления в разработке высокопроизводительных средств вычислений – универсальных однородных вычислительных систем с программируемой структурой (УВС).

Если прообразом ЭВМ является вычислитель-одиночка, то прообраз УВС – коллектив вычислителей, сообща работающих над одной сложной задачей.

УВС состоит из совокупности элементарных машин, которые представляют собой одинаковые универсальные машины, объединенные между собой программно управляемой системой коммутаций.

Это направление получило дальнейшее развитие как в Институте математики Сибирского отдел.АН СССР,так и в ряде организаций нашей страны. Здесь следует назвать работы Э.В. Евреинова, М.А. Карцева, А.А. Папернова, В.Б. Смолова, Е.П. Балашова, Д.А. Поспелова, С.Д. Пашкеева, И.Я. Акушского, Д.И. Юдицкого и других. Сейчас это направление стало довольно популярным и в его разработку включились многие коллективы.

Близким к данному направлению является разработка вычислительных сред. В этой области ведут работы А.В. Каляев, Г.Е. Пухов, Э.В. Евреинов, И.В. Прангишвили, В.Г. Лазарев и другие.

Работы по созданию вычислительных систем высокой производительности, основывающиеся на параллельной обработке информации, начаты в США и в других странах.

В ходе работ над УВС выяснилось, что применение упомянутых выше принципов существенно облегчает разработку математического обеспечения. Благодаря этому и второе препятствие на пути создания УВС оказалось преодолимым.

В предлагаемой диссертации решаются проблемы, связанные с синтезом функциональных структур вычислительных систем с программно управляемыми связями.

Цель работы – создание основ построения и исследования надежных, простых в технической и математической эксплуатации высокопроизводительных средств переработки информации, предъявляющих минимальные требования к физико-технологической элементной базе и к системе математического обеспечения.

Предмет исследования – синтез функциональных структур технических средств – универсальных однородных вычислительных систем с програмно управляемыми связями; синтез структурных моделей процесса решения сложных задач; разработка математического аппарата для отображения структур –

ных моделей задач на функциональную структуру вычислительной системы.

Метод исследования - сочетание теоретических и экспериментальных разработок при широком использовании методов структурного моделирования, подобных развитым в аналоговой вычислительной технике. В отличие от последних для высокопроизводительных вычислительных систем требуется строить более сложные структурные модели с большим числом модулей и разрабатывать оптимальные или близкие к ним методы решения сложных задач.

Диссертация состоит из пяти глав:

В первой главе изложены принципы параллельности, однородности и программной изменяемости структуры, а также сформулированы общие требования к УВС.

Во второй главе излагаются основы синтеза функциональных структур УВС.

В третьей главе исследуются общие свойства параллельных алгоритмов с целью разработки основ методов синтеза структурных моделей задач, реализуемых на УВС, и методов отображения этих моделей на функциональную структуру УВС (математическое обеспечение для УВС).

В четвертой главе исследуются структурные модели процесса решения разнообразных классов задач с целью выявления возможности их эффективной реализации на УВС.

В пятой главе обсуждается структура и особенности разработки математического обеспечения для УВС.

ГЛАВА I. ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА ВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ

I. Каждая задача имеет практический смысл, если время ее решения может быть сделано меньше некоторой величины $t_{пр}$ (например, прогнозирование какого-либо события должно быть выполнено раньше, чем произойдет само событие и т.п.).

Величине $t_{пр}$ можно поставить в соответствие предельное количество операций $R_{пр}$, которое может быть выполнено за время $t_{пр}$ при наибольшей величине производительности $P_{посл}$, достигнутой на вычислительных устройствах последовательного типа.

Величина $R_{пр}$ служит наглядным показателем практической реализуемости алгоритма. Алгоритм реализуем, если

$$h_{посл} < R_{пр}, \tag{1.1}$$

где $h_{посл}$ – длина алгоритма, измеряемая количеством последовательно выполняемых операций.

В соответствии с принципом параллельности процесс решения задачи представляется в виде параллельных ветвей вычислений. В этом случае длина параллельного алгоритма $h_{пар}$ определяется числом операций т о л ь к о в самой длинной из ветвей, поэтому использование принципа параллельности позволяет расширить круг практически реализуемых задач при достигнутом уровне производительности $P_{посл}$.

Как показало исследование большого числа классов задач [2, 6], справедливо

$$h_{пар} \ll h_{посл}. \tag{1.2}$$

При этом как для $h_{посл} < R_{пр}$, так и для $h_{посл} > R_{пр}$ имеет место

$$h_{пар} \ll R_{пр}, \tag{1.3}$$

то есть многие классы задач оказываются не только параллельно реализуемыми, но для них открывается возможность широкого маневра при выборе рабочей частоты элементов вычислительных устройств.

Тем самым существенно облегчается задача построения вычислительных средств высокой производительности и, более того, становится реальным построение средств вычислений с производительностью порядка $10^8 - 10^9$ опер/сек на существующей технологической базе.

2. Принцип параллельности открывает принципиальную возможность построения высокопроизводительных средств вычислений на уже имеющихся типах элементов. Для реализации этой возможности нужно изготовить огромное число элементов, собрать из них устройства, объединить эти устройства в систему и обеспечить ее работоспособность.

Ясно, что для успешного решения этой проблемы необходимо положить в основу разработки такие структурные схемы,которые в наибольшей степени ослабили бы требования к технологическому процессу.

С этой целью в основу разработки УВС был положен принцип однородности, который означает, что УВС строится из одинаковых и одинаковым образом соединенных вычислительных модулей. Это позволяет свести технологический процесс к многократному изготовлению сравнительно простого вычислительного модуля.При этом оказалось, что однородность структуры существенно облегчает также и создание математического обеспечения.

3. Использование принципа программной изменяемости структуры позволяет создавать УВС, сочетающие достоинства универсальных и проблемно-ориентированных средств вычислений.Подобно первым,УВС не предъявляют высоких требований к срокам разработки и изготовления, подобно структуре вторых, структура УВС хорошо согласуется со структурными схемами решаемых задач,что существенно облегчает разработку математического обеспечения.

/

ГЛАВА 2. ОДНОРОДНЫЕ УНИВЕРСАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С УПРАВЛЯЕМЫМИ СВЯЗЯМИ

В данной главе рассматриваются свойства универсальных однородных вычислительных систем с программно управляемыми связями (УВС) - одного из наиболее перспективных направлений создания высокопроизводительных средств переработки информации.

I. УВС представляет собой совокупность одинаковых и одинаково соединенных друг с другом элементарных машин (вычислительных модулей). Каждая элементарная машина (ЭМ) состоит из двух конечных автоматов: функционального и коммутационно-настроечного.

Функциональный автомат представляет собой программный автомат Глушкова с конечным объемом памяти и универсальным набором операций, дополненным набором системных операций (П,Пр,ОУП, ОБП, Н).

Операция П состоит в п е р е д а ч е в выходной канал группы кодов, хранящихся в памяти автомата, а операция Пр - в п р и е м е группы кодов из входного канала и ее записи в память.

Операция ОУП (обобщенного условного перехода)передает управление очередной команде или команде с заданным адресом в зависимости от выполнения некоторого обобщенного условия (функции от условий, вырабатываемых в ЭМ, выделенных с помощью операции Н).

Операция ОБП(обобщенного безусловного перехода) посылает команду, которую выполняют все ЭМ, выделенные с помощью операции Н.

Операция Н (настройки) определяет разбиение системы на подсистемы и выделяет множества ЭМ, участвующих в выполнении операций ОУП и ОБП.

Коммутационно-настроечный автомат включает в себя коммутаторы и каналы связи, а также устройство управления,которое в зависимости от своего состояния и команды системы, поступившей на его вход из функционального автомата, реализует те или иные схемы взаимодействия с остальными ЭМ системы.

2. Для УВС высокой производительности необходимо учитывать возможность изменения в процессе ее функционирования чис-

ла вычислительных модулей (либо при выходе их из строя, либо при наращивании УВС). Важно, чтобы при этом сохранялось свойство однородности [25] .

В диссертации вводится определение однородности коммутаций (К-однородность). Пусть M_1 ,..., M_S – классы конечного множества $M = \bigcup_{i=1}^{S} M_i = M$, $M_i \cap M_j = 0$, $i \neq j$.

Бинарное отношение $R = R_{ij} \subset M_i \times M_j$ назовем з а м к н у т ы м , если хотя бы при одном порядке нумерации элементов в классах для всех $a_{in} \in M_i$ и $a_{jm} \in M_j$, связанных отношением R , имеет место

$$R(a_{in}, a_{jm}) \longrightarrow R(a_{in \oplus 1}, a_{jm \oplus 1}), \qquad (2.1)$$

где знак $\oplus$ означает сложение по модулям $m_i = |M_i|$ и m_j = = M_j , соответственно.

Множество M , на котором заданы замкнутые бинарные отношения R_1 , R_2 ,..., R_r , назовем к-однородными по R_1 , R_2 ,..., R_r, или просто К-о д н о р о д н ы м и , если других отношений нет.

К-однородное множество, которое остается К-однородным после удаления из него любого элемента, назовем УК - о д н о р о д н ы м .

ТЕОРЕМА. К - о д н о р о д н о е м н о ж е с т в о M, с о с т о я щ е е и з к л а с с о в M_1 ,..., M_S , У К - о д н о р о д н о п о R т о г д а и т о л ь к о т о г д а, к о г д а о т н о ш е н и е R (a_{in} , a_{jm}) с у щ е с т в у е т д л я в с е х п а р э л е м е н т о в и з к л а с с о в M_i и M_j , н а к о т о р ы х э т о о т н о ш е н и е о п р е д е л е н о.

ЛЕММА. УК-однородные множества и после удаления из них любого элемента остаются УК-однородными, если в каждом из их классов остается более одного элемента.

3. В диссертации вводятся и классифицируются схемы обмена. Под схемой обмена понимается ориентированный граф, вершины которого соответствуют ЭМ, а вершины, соединенные дугой,- парам ЭМ, обменивающимся в данный момент времени информацией.

Совокупность всех схем обмена, встречающихся при реализации данного алгоритма, оьразует п о л н у ю с х е м у о б м е н а .

Важно выделить схемы обмена, которые обладают достаточной общностью и в то же время не предъявляют высоких требований к системе коммутаций. Предлагается один из классов таких схем, названных п р о с т ы м и, к неизолированным вершинам которых примыкает либо одна дуга, либо несколько когерентных дуг (т.е. таких, которым соответствует одновременная передача одной и той же информации).

Среди простых схем важную роль играют два типа, которые сравнительно просто могут быть реализованы однотактно:

т р а н с л я ц и о н н ы е схемы, у которых из одной вершины выходят когерентные дуги ко всем остальным неизолированным вершинам;

п а р н ы е с м е ж н ы е схемы, у которых дугами соединяются только соседние по номеру вершины.

Эти типы схем обмена встречаются при реализации на УВС почти всех задач либо сами по себе, либо как составляющие более сложных схем обмена, реализуемых однотактно (см.табл.4.2). В частности, распространенная схема обмена, изображаемая полным графом, реализуется n - тактным повторением трансляционной схемы, где n - число вершин в графе. Такая многотактная схема обмена названа ц и к л и ч е с к о й т р а н с л я ц и о н н о й.

4. Изучение схем обмена и относительных затрат времени на их реализацию в различных задачах позволило приступить к целенаправленному синтезу функциональных структур УВС, устойчивых к изменению числа элементов.

Одним из основных результатов этой работы явилась разработка и исследование структур магистрального типа. Простейший пример таких структур - мономагистральные УВС. У них имеется магистральный канал, к которому через свои коммутаторы подключены все ЭМ.

Мономагистральные УВС позволяют однотактно реализовать основные схемы обмена: трансляционные и парные смежные. УВС может быть легко программно разбита на независимые подсистемы. Технически просто происходит наращивание УВС для этого нужно только включить в магистральный канал дополнительные коммутаторы по числу добавляемых ЭМ.

Нетрудно видеть, что, если рассматривать магистральный канал как одноэлементное множество, а все ЭМ - как другое множество, то удовлетворяется условие К-однородности (2.I). Это условие остается в силе и при изменении числа ЭМ, т.е. мономагистральные УВС УК-однородны относительно ЭМ.

Указанные свойства, а также простота и высокая надежность магистрального канала позволяют успешно применять такие структуры на практике.

5. Примером мономагистральной УВС может служить УВС "Минск-222", разработанная в 1966 г. по предложению и при участии автора Институтом математики СО АН СССР и Минским проектным бюро завода им. Орджоникидзе [II].

В качестве ЭМ служила серийная ЭВМ "Минск-22", к которой был добавлен блок системы объемом около 80 стандартных элементов, что составляет менее 1,5% от объема оборудования арифметического устройства и устройства управления ЭВМ "Минск-22". В этот блок входят схемы, реализующие команды системы, коммутатор и трехразрядный регистр настройки (РН), управляющий изменением структуры информационных и управляющих связей в системе. Оказалось, что при выбранном техническом варианте УВС затраты на системные взаимодействия составляют лишь несколько процентов от общего времени счета (см. гл. 4).

6. 4-летний опыт математической и технической эксплуатации системы "Минск-222" полностью подтвердил правильность теоретических исследований автора и показал эффективность и жизнеспособность таких систем. Полученный опыт послужил основой для разработки многомашинных УВС в ряде организаций страны.

При увеличении числа ЭМ в УВС растет вероятность выхода из строя магистрального канала. Поэтому для разрабатываемых в настоящее время УВС с числом ЭМ порядка 10^1, 10^2 и более в диссертации предложены два основных типа полностью УК-однородных структур: мультимагистральные УВС и магистральные сети. Первые состоят из нескольких не соединенных друг с другом магистралей, к каждой из которых присоединены все ЭМ. У вторых магистрали образуют n - мерную решетку, в узлах которой находятся коммутаторы, соединяющие между собой магистрали различных измерений, причем каждый такой коммутатор соединен с одной из ЭМ.

Данные структуры положены в основу разработки трех проблемно-ориентированных УВС для решения некоторых специальных классов задач [26–28].

7. Как правило, команды ЭВМ основываются на принципе адресности. С помощью адреса задаются обращения к ячейкам оперативной памяти, к массивам, находящимся на вспомогательных памятях, а также к различным внешним устройствам. Таким же способом можно построить и системные команды. Однако, когда число машин велико и приходится учитывать возможность их выхода из строя в процессе решения задачи, непосредственное применение адресуемости ЭМ встречает некоторые трудности. Задание списка адресов ЭМ, которым предназначается та или иная информация, неудобно из-за больших размеров списка, а также необходимости вносить в него изменения при выходе ЭМ из строя и передаче их функций другим ЭМ. Нами была рассмотрена одна из возможностей преодоления указанных трудностей путем отказа от адресуемости ЭМ.

Определенный шаг в этом направлении был сделан при разработке системы "Минск-222". В этой системе обмен информацией между ЭМ и организация взаимодействия ЭМ для управления процессом вычислений выполняются без указания адресов ЭМ.

Обмен осуществляется с помощью предварительной синхронизации участвующих в нем ЭМ и команд "передача в магистральный канал" и " прием из магистрального канала".

В командах управления безадресность достигается путем введения у каждой ЭМ состояний модальности, которые изменяются специальными командами настройки, благодаря чему в выполнении команд управления ОУП и ОБП участвуют ЭМ, находящиеся в определенных состояниях модальности.

8. В диссертации рассмотрен вопрос о возможности использования в УВС ЭМ, у которых длительности выполнения арифметических операций – случайная функция от операнд. С этой целью были проделаны теоретико-вероятностные исследования относительных простоев ЭМ в зависимости от числа ЭМ в УВС ℓ и количества операций τ, выполняемых между двумя последовательными синхронизациями.

Пусть средняя длительность операций в i – й ЭМ равна

$$T_{\tau i} = \frac{1}{\tau} \sum_{j=1}^{\tau} t_{ji} \quad , \qquad (2.2)$$

где t_{ji} - время выполнения j - й по счету операции в i - й ЭМ.

Интересующие нас потери из-за десинхронизации равны

$$\tau_{z\ell} = \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\overset{\infty}{T}_{z\ell} - T_{zi} \right), \qquad (2.3)$$

где $\overset{\infty}{T}_{z\ell} = \max\limits_{1 \le i \le \ell} T_{zi}$.

Из цепочки неравенств

$$M\tau_{z\ell}^2 \le \sum_{h=1}^{\ell} P\left(\overset{\infty}{T}_{z\ell} = T_{zh}\right) M\tau_{z\ell}^2 \Big|_{\overset{\infty}{T}_{z\ell} = T_{zh}} \le \sum_{h=1}^{\ell} \frac{1}{\ell^2} M\left(\sum_{i=1}^{\ell} (T_{zh} - T_{zi})\right)^2 \qquad (2.4)$$

следует оценка сверху для математического ожидания

$$M\tau_{z\ell} \le \sqrt{\frac{\ell-1}{z^2} \sum^{s} z_K D\tau_K} \qquad (2.5)$$

и для относительных простоев ЭМ

$$\theta = \frac{M\tau_z \ell}{\frac{1}{z}\sum^{s} z_K M\tau_K} \le \sqrt{(\ell-1)\frac{\sum_{K=1} z_K D\tau_K}{\left(\sum_{K=1} z_K M\tau_K\right)^2}} . \qquad (2.6)$$

Оценка (2.6) зависит от конкретного алгоритма, поэтому часто удобнее пользоваться более грубой, но зато абсолютной оценкой

$$\theta_{abc} = A_{abc} \sqrt{\frac{\ell-1}{z}}, \qquad (2.7)$$

где $A_{abc} = \sqrt{\dfrac{\max\limits_{K} D\tau_K}{(\min\limits_{K} M\tau_K)^2}}$.

Определение $M\tau_K$ и $D\tau_K$ сопряжено со сложными теоретико-вероятностными расчетами. В диссертации приводится соответствующая методика их определения для различных типов команд. Для системы команд "Минск-22" оказалось, что A_{abc} = 0,35. Это значение, соответствующее последовательности команд типа сложения с плавающей запятой, примерно в 7 раз больше, чем для последовательности из команд умножения.

Для некоторых типичных случаев можно воспользоваться более точной оценкой (2.6). Например, для многих задач линейной алгебры основное время счета уходит на вычисление сум парных произведений. Для "Минск-222" оценка по формуле (2.6) оказывается равной

$$\tilde{\theta} = 0{,}22\sqrt{\frac{\ell-1}{\tau}} . \tag{2.8}$$

Как показывает анализ задач (см.гл. 4), обычно $\tau \ll \ell$ и лишь при $\ell = 10^2-10^3$ τ может стать сравнимым или меньше ℓ. Из приведенных оценок можно видеть, что для УВС малой и средней производительности ($\ell < 100$) можно применять команды, длительность которых является случайной функцией операнд. При увеличении числа ЭМ желательно применять команды, длительность которых мало зависит от операнд. При этом, в первую очередь нужно сделать постоянной длительность операций типа сложения с плавающей запятой.

ГЛАВА 3. ОБЩИЕ СВОЙСТВА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

I. В соответствии с принципом параллельности возникла необходимость в разработке методов, которые позволяют эффективно и в то же время сравнительно просто распределять операции данного алгоритма между большим числом ЭМ.

В этой области имелись лишь некоторые результаты, полученные в связи с мультипрограммированием для ЭВМ с несколькими процессорами. Все известные методы такого рода основывались на локальных связях между операциями и оказались пригодными лишь для сравнительно небольшого числа процессоров.

В данной главе описывается и обосновывается один из путей решения данной проблемы для большого числа вычислительных модулей (процессоров с памятью). В отличие от методов, применяющихся в мультипрограммировании, предлагаемый путь исходит не из локальных связей между операциями, а из наиболее общей структурной модели (архитектуры) алгоритма.

В связи с этим пришлось ввести и исследовать новый класс алгоритмов - параллельные алгоритмы (р-алгоритмы) [2,7].

Приведем основные определения, которые для краткости дадим в бэкусовой форме, расширив ее метасимволом < ‖ > ,означающим, что соединенные им символы выполняются одновременно и независимо друг от друга.

<р-алгоритм> ::= <алгоритм> | <р-кортеж> ,
<р-кортеж> ::= <р-операция> | <р-кортеж> <р-операция> ,
<р-операция> ::= <операция системы> ‖ <р-операция> ,
<операция> ::= <операция ЭВМ> | <операция системы> ,
<операция системы> ::= <П> | <Пр> | <ОУП> | <ОБП> | <Н> ,
<ОУП> ::= <ОУП> | <синхронизация> ,
<Н> ::= <Н> | <самонастройка> ,
<обмен> ::= <передача> ‖ <р-прием> ,
<р-прием> ::= <прием> ‖ <р-прием> ,
<р-синхр> ::= <синхр> ‖ <р-синхр> ,
<р-ОУП> ::= <ОУП> ‖ <р-ОУП> ,
<р-ОБП> ::= <ОБП> ‖ <р-операция> ,
<р-Н> ::= <настройка> ‖ <р-операция> .

Далее под р-кортежем будем понимать прямоугольный р-кортеж, в котором все р-операции одинаковой высоты L и хотя бы одна р-операция не содержит пустых операций.

Множество прямоугольных р-кортежей, с помощью которых может быть представлен данный р-алгоритм, образует с е м е й с т в о р-кортежей. Р-кортежи одного семейства неразличимы по составу операций, но могут отличаться числом пустых операций Λ и числом операций системы C.

Введем коэффициенты эффективности, характеризующие качество р-кортежей,

$$\Delta_\Lambda = \frac{\Lambda}{\chi}\ , \ \Delta_C = \frac{C}{\chi}\ , \ \Delta = \Delta_\Lambda + \Delta_C = \frac{\Lambda + C}{\chi}\ , \qquad (3.1)$$

где χ - общее число операций (объем) последовательного алгоритма из данного семейства,

и

$$\delta_\Lambda = \frac{1}{1+\Delta_C}\ , \ \delta_C = \frac{1}{1+\Delta_C}\ , \quad \delta = \frac{1}{1+\Delta}\ . \qquad (3.2)$$

Перенумеруем операциии в каждой р-операции р-кортежа от 1 до L.

Последовательность операций системы с одинаковыми номерами, расположенных в том же порядке, что и в прямоугольном р-кортеже, назовем в е т в ь ю.

Каждую из исходных операнд припишем одной из ветвей. Очевидно, что если нужные для данной ветви операнды находятся или вычисляются в другой ветви, то в эти ветви должны быть вставлены соответствующие операции системы.

2. Задача заключается в том, чтобы для заданной ширины р-кортежа найти оптимальное размещение операций системы между ветвями, которому соответствует наибольшее значение δ. Естественно, что весьма важно найти решение этой задачи в наиболее общем виде. С этой целью было предпринято исследование свойств р-алгоритмов и, прежде всего, методов их преобразования.

В ходе этих исследований была введена и изучена характе -

ристическая функция (х.ф.), устанавливающая зависимость между длиной и шириной р-кортежей одного семейства.

Доказаны следующие свойства х.ф.

ТЕОРЕМА 3.1. Характеристическая функция представляет собой монотонную последовательность точек на плоскости h, L

$$a_1(h_1,L_1)=a_m(h_m,\hat{L}_m),\ldots,a_i(h_i,L_i),$$

$$a_{i+1}(h_{i+1},L_{i+1}),\ldots,a_n(h_n,L_n)=a_{np}(h_{np},1),$$

где $h_{i+1}\geq h_i$, $L_{i+1}\leq L_i$, точка a_{np} (h_{np}, 1) соответствует последовательному алгоритму, точка $a_m(h_m\hat{L}_m)$ – р-кортежу с минимальной длиной.

ЛЕММА 3.1. Любая точка, не принадлежащая х.ф., соответствует р-кортежу, коэффициент эффективности δ которого может быть увеличен путем уменьшения L при том же h, либо путем уменьшения h при том же L.

ЛЕММА 3.2. Х.ф. состоит из двух областей: эффективной $1\leq L\leq L_g$ и неэффективной $L_g<L\leq\hat{L}_m$.

ЛЕММА 3.3. Любую р-операцию высотой ℓ, кратной L, можно заменить последовательностью из ℓ/L р-операций, составленных из тех же операций, что и исходная. Назовем этот процесс L-разверткой.

СЛЕДСТВИЕ. Если ℓ не кратно L, то с помощью L-развертки получаем последовательно из $entier\,(\ell/L)+1$ р-операций, в которой содержится λ пустых операций, где λ - дополнение ℓ до ближайшего числа, кратного L.

ТЕОРЕМА 3.2. Если существует общий делитель L чисел непустых операций ℓ_z в р-операциях, то данный р-кортеж может быть преобразован в эффективный.

СЛЕДСТВИЕ. Если известна эффективная точка р-алгоритма $e(h_e^*, L_e^*)$ и $L_e^*=L_aL_c$, где L_a и L_c - целые, то на х.ф. существуют эффективные точки $a(h_a=L_ch_e^*L_a)$ и $c(h_c=L_ah_e^*,L_c)$.

3. Описанная выше методика построения схемы р-алгоритма основана на знании эффективной схемы для базовой точки. В диссертации доказаны некоторые свойства х.ф., которые облегчают поиск базовых точек. Эта процедура, однако, остается сравнительно сложной, поэтому были приложены усилия для отыскания простой методики нахождения схем для базовых точек.

Автором была предложена методика распараллеливания по циклам [9]. Идея ее заключается в следующем. В качестве исходной берется схема последовательного алгоритма. В этой схеме находятся циклы, обладающие следующими свойствами: независимостью (все повторения циклов могут выполняться независимо друг от друга), достаточно большим числом повторений n и тем, что они охватывают в совокупности основные (по времени реализации) операторы. Тогда схема для базовой точки $L_б = n$ представляется в виде n одинаковых ветвей. Каждая из ветвей изображает собой последовательную схему, в которую введены операторы системы для организации взаимодействия между ветвями. Исходные операнды при этом приписываются к соответствующим ветвям.

4. Кроме указанной схемы распараллеливания, наличие независимых (или слабо зависимых) циклов позволяет создать и другой эффективный способ распределения операций между ветвями вычислений. Эта схема, названная конвейерной, предусматривает разделение процесса вычислений, вообще говоря, на различные ветви вычислений. Операторы, охватываемые циклом, по возможности равномерно распределяются между ветвями. Каждая ветвь i, кроме первой и последней, получает операнды от ветви $i-1$, перерабатывает их и посылает результаты в ветвь $i+1$. У первой ветви операндами служат исходные данные, последняя ветвь выдает результаты. Показаны условия, при которых конвейерная схема оказывается эффективной.

Кроме этих двух схем, может применяться смешанная схема, представляющая ту или иную комбинацию основной и конвейерной.

5. С переходом к параллельным алгоритмам возникает вопрос о языке. Обычные языки, развитые для описания последовательных процессов, непосредственно применить нельзя. Необходим язык, который описывал бы процесс вычислений не только во времени, но и в пространстве. Желательно, чтобы этот язык был возможно ближе к известным языкам.

С этой целью предлагается матричный р-язык, который в зависимости от ситуации может использоваться в одной из трех форм:

а) в виде прямоугольной матрицы, строки которой представляют ветви р-кортежа, а столбцы - р-операторы. Взаимодействия между ветвями выражаются в этом языке с помощью операторов системы, соответствующих приведенным выше операциям системы;

б) последовательности векторов, каждый из которых состоит из операторов системы, соответствующих столбцу в матричной форме;

в) граф-схемы, в которой операторы изображаются точками, а зависимости между ними - дугами.

ГЛАВА 4. ИССЛЕДОВАНИЕ СТРУКТУРНЫХ МОДЕЛЕЙ ПРОЦЕССА РЕШЕНИЯ ЗАДАЧ

В данной главе исследуются основные классы вычислительных задач. Главные цели данного исследования – установить, насколько широк круг параллельно реализуемых задач; выявить основные типы структурных схем р-алгоритмов и установить требования к функциональным схемам; оценить значение эффективности для основных классов задач; определить, насколько широк круг задач, для которых пригодна методика распараллеливания по циклам; оценить затраты на программирование; отработать методы решения задач на УВС.

Исследование задач проводилось путем составления схем р-алгоритмов, программирования для системы "Минск-222", просчетов на системе и на одиночной ЭВМ при одновременном хронометраже с помощью специальных анализирующих программ как общего времени решения, так и затрат на обмен, синхронизацию и т.п. [18]

Исследованию были подвергнуты задачи различных классов (табл. 4.I). Остановимся на нескольких из них.

I. Решение системы линейных уравнений методом простой итерации [12]. Решение этой задачи сводится к вычислению новых значений переменных

$$x_i^{(k)} = g_i + \sum_{j=1}^{n} b_{ij}\, x_j^{(k-1)} \quad , \quad i = \overline{1,n} \; , \tag{4.1}$$

пока для всех x_i не будет достигнута нужная точность

$$\left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon \; . \tag{4.2}$$

Операторная схема соответствующего р-алгоритма состоит из ℓ одинаковых ветвей, к каждой из которых отнесено r строк матрицы (b_{ij}).

$$\dot{A}_1^{jik} \dot{P}_1(j)\, \dot{A}_2^{ik} \dot{P}_2(i)\, \dot{O}^{k} \dot{P}_3(k)\, \dot{q} \; , \tag{4.3}$$

где

$$r = entier\,((n-1/2)/\ell)) + 1 \tag{4.4}$$

а р-операторы выполняют следующие функции:

$\dot A_1^{ijk}$ и $\dot A_2^{ik}$ вычисляют выражения (4.1) и (4.2);
$\dot P_1(j)$, $\dot P_2(i)$ организуют циклы внутри каждой ветви;
$\dot P_3(k)$ – ОУП, который в зависимости от условия (4.2) продолжает счет или передает управление р-оператору конца $Я$;
$\dot O^k$ осуществляет циклическую трансляционную схему обмена для пересылки $x_1^{(k)}, \ldots, x_n^{(k)}$.

2. Обращение матриц методом пополнения [6]. Эта задача сводится к последовательному вычислению выражений:

$$\alpha_{ij}^{(k)} = \alpha_{ij}^{(k-1)} - \frac{e_j^{(k)}}{1 + e_k^{(k)}} \alpha_{ik}^{(k-1)}, \tag{4.5}$$

где

$$e_j^{(k)} = z_{kj} + \sum_{i=1}^{k-1} w_{ki} \alpha_{ij}^{(k-1)}, \tag{4.6}$$

$$z_{kj} = \text{если } j \geq k, \text{ тогда } w_{kj}, \text{ иначе } 0; \tag{4.7}$$
$k = \overline{1,n}$; $i = \overline{1,k}$; $j = \overline{1,n}$.

w_{ij} – элементы исходной матрицы, из которой вычтена единичная.

$\alpha_{ij}^{(0)}$ – элементы единичной матрицы.

Схема р-алгоритма состоит из ℓ одинаковых ветвей, к каждой из которых отнесено r столбцов матрицы (α_{ij}) и r строк матрицы (w_{ij}),

$$\dot O_1^k \dot A_1^{ijk} \dot P_1(i) \dot P_2(j) \dot A_2^k \dot O_2^k \dot A_3^{jk} \dot A_4^{ijk} \dot P_3(i) \dot P_4(j) \dot P_5(k) \dot Я, \qquad i \le \ell-1,\ j < r,\ k < n,\ i < k,\ j < r \tag{4.8}$$

где r то же, что в (4.4), р-операторы $\dot A_1^{ijk}$, $\dot A_2^k$, $\dot A_3^{jk}$, $\dot A_4^{ijk}$ вычисляют соответственно $e_j^{(k)}$, $1/(1+e_k^{(k)})$, $e_j^{(k)}/(1+e_k^{(k)})$, $\alpha_{ij}^{(k)}$. При этом у р-оператора $\dot A_2^k$ $\ell-1$ составляющих пустые, а у остальных трех все составляющие одинаковы;

Р-операторы $\dot P_1(i)$, $\dot P_2(j)$, $\dot P_3(i)$, $\dot P_4(j)$, $\dot P_5(k)$ организуют счет по соответствующим циклам в каждой из ветвей. Р-операторы $\dot O_1^k$, $\dot O_2^k$, $\dot O_3^k$ организуют трансляционные схемы обмена для передачи $w_{k1}, \ldots, w_{kn}$; $1/(1+e_k^{(k)})$; $\alpha_{1k}^{(k-1)}, \ldots, \alpha_{kk}^{(k-1)}$, соответственно.

3. Вычисление собственных значений матрицы по методу Данилевского. Идея метода состоит в следующем. Данная матрица A,

рассматриваемая как матрица оператора в базисе $e_1 = (1,0,\ldots,0)'$, $e_2 = (0,1,\ldots,0)$, ..., $e_n = (0,0,\ldots,1)$, за $n-1$ шагов преобразуется к базису e_1, Ae_1, ..., $A^{n-1}e_1$. Каждый шаг состоит в переходе от базиса e_1, Ae_1, ..., $A^{k-1}e_1$, e_{k+1}, ..., e_n к базису e_1, Ae_1, ..., $A^{k-1}e_1$, $A^k e_1$, ..., e_{k+2}, ..., e_n.

При этом используются следующие формулы:

$$b^{(k)}_{k+1,j} = a^{(k)}_{k+1,j} / a^{(k)}_{k+1,k} , \tag{4.9}$$

$$b^{(k)}_{ij} = a^{(k)}_{ij} - a^{(k)}_{ik}\, b^{(k)}_{k+1,j} \qquad (i \neq k+1), \tag{4.10}$$

$$a^{(k+1)}_{ij} = b^{(k)}_{ij} \qquad (j \neq k+1), \tag{4.11}$$

$$a^{(k+1)}_{i,k+1} = \sum_{j=1}^{n} b^{(k)}_{ij}\, a^{(k)}_{jk} . \tag{4.12}$$

Здесь $a^{(k)}_{ij}$ и $b^{(k)}_{ij}$ – элементы вспомогательных матриц $A^{(k)}$ и $B^{(k)}$, $A^{(1)} = A$.

В схеме р-алгоритма отнесем к каждой из ℓ ветвей τ строк матрицы A^k.

$$\underbrace{\dot{O}_1^{k}\, \dot{A}_1^{kj}\, \dot{O}_2^{k}\, \underbrace{\underbrace{\dot{A}_2^{kij}\, \dot{P}_1(j)}_{j<n}\, \dot{P}_2(i)}_{i<\tau}\, \dot{P}_3(k)}_{k<n-1}\, \dot{Я} , \tag{4.13}$$

где τ то же, что и в (4.4), р-операторы $\dot{A}_1^{kj}$ и $\dot{A}_2^{kij}$ вычисляют выражения (4.9) и (4.10) – (4.12). При этом р-оператор $\dot{A}_1^{kj}$ содержит $\ell - 1$ пустых составляющих, все составляющие $\dot{A}_2^{kij}$ одинаковы;

$\dot{O}_1^{k}$ реализует циклическую трансляционную схему обмена для пересылки элементов $a^{(k)}_{k}$, ..., $a^{(k)}_{nk}$;

$\dot{O}_2^{k}$ осуществляет трансляционную схему обмена для пересылки во все ветви групп элементов $b^{(k)}_{k+1,k+1}$, ..., $b^{(k)}_{k+1,n}$;

$\dot{P}_1(j)$, $\dot{P}_2(i)$, $\dot{P}_3(k)$ организуют счет по соответствующим циклам в каждой ветви.

4. Решение систем обыкновенных дифференциальных уравнений методом Рунге-Кутта [6]. Система дифференциальных уравнений,

приведенная к нормальной форме

$$
\begin{aligned}
y_1' &= f_1(y_1, \dots, y_n) \;, \\
&\dots\dots\dots\dots\dots \;; \\
y_n' &= f_n(y_1, \dots, y_n) \;,
\end{aligned}
$$

согласно методу Рунге-Кутта решается последовательными итера- циями. На каждой S - й итерации вычисляются

$$
\begin{aligned}
K^s_{1i} &= h f_i(y^s_1, \dots, y^s_n) \;, \\
K^s_{ji} &= h f_i(Z^s_{j1}, \dots, Z^s_{jn}) \;, \qquad\qquad (4.14) \\
Z^s_{ji} &= y^s_i \quad K^s_{j-1,i} \;, \\
j &= 2,3,4; \quad i = \overline{1,n} \;,
\end{aligned}
$$

где h - половина шага интегрирования, и

$$\Delta y^s_i = \frac{1}{3}\left(K^s_{1i} + 2K^s_i + 2K^s_{3i}\right), \quad i = \overline{1,n} \;; \qquad (4.15)$$

$$y^{s+1}_i = y^s_i + \Delta y^s_i \;. \qquad (4.16)$$

Пусть каждая из ветвей хранит и вычисляет переменные, соответ- ствующие значениям параметра i .

$$\underbrace{\dot O^s \underbrace{\underbrace{\dot A^{sij}}_{j<n} \dot P_1(j)}_{i<z} \dot P_2(i)}_{s} \dot P_3(s)\, \dot\vartheta \;. \qquad (4.17)$$

Здесь z то же, что и в (4.4), функции р-операторов сле- дующие:

$\dot O^s$ реализует циклическую трансляционную схему обмена для пересылки кодов $y^s_i, \dots, y^s_n$ и $Z^s_{j1}, \dots, Z^s_{jn}$; $\dot A^{kij}$ вычисляет выражения (4.14)-(4.16); $\dot P_1(j)$, $\dot P_2(i)$, $\dot P_3(s)$ орга- низуют вычисления по соответствующим циклам в каждой из ветвей.

5. Решение дифференциальных уравнений в частных производных. Рассмотрим для примера решение уравнения Пуассона [6]

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x,y) \tag{4.18}$$

для прямоугольной области D . Разобьем область D на квадраты с помощью сетки с шагом h . Для вычисления значений функции в узловой точке (j , k) на s – м последовательном приближении воспользуемся формулой

$$U^{s}_{jk} = \frac{1}{4}\left(U^{s-1}_{j-1,k} + U^{s-1}_{j+1,k} + U^{s-1}_{j,k-1} + U^{s-1}_{j,k-1} - h^2 g_{jk}\right). \tag{4.19}$$

Процесс продолжается, пока для всех точек не будут удовлетворены условия

$$\left| U^{s}_{jk} - U^{s-1}_{jk} \right| \leq \varepsilon . \tag{4.20}$$

Из формул (4.19), (4.20) видно, что можно одновременно вычислять все N точек. N обычно велико, поэтому для числа ветвей ℓ всегда на практике можно считать справедливым соотношение $\ell \ll N$.

Разобьем область D на части прямыми, параллельными одной из сторон области (лучше меньшей), так, чтобы в каждой части было число точек, примерно равное N/ℓ , и пусть каждая ветвь вычисляет точки в одной из частей.

Соответствующую схему р–алгоритма можно представить так

$$\dot{O}^{s}\, \dot{A}^{skj}\, \dot{P}_1(k)\, \dot{P}_2(j)\, \dot{P}_3(s)\, \dot{Я} , \tag{4.21}$$

$k < m$, $j < r$, s

где r то же, что и в (4.4), а функции р-операторов следующие:

$\dot{O}^{s}$ реализует схему двустороннего парного смежного обмена m значениями функции U^{s-1}_{jk} в пограничных точках;

$\dot{A}^{skj}$ вычисляет выражения (4.19), (4.20);

$\dot{P}_1(k)$, $\dot{P}_2(j)$ организуют циклы по k и по j в каждой из ветвей;

$\dot{P}_3(s)$ – р-оператор ОУП, устанавливающий конец счета при выполнении условия (4.20) во всех ветвях.

6. Применение метода прогонки для решения дифференциальных уравнений в частных производных [15]. Метод прогонки является одним из наиболее распространенных при решении дифференциальных уравнений в частных производных и особенно их систем.

В диссертации рассмотрены два примера задач такого рода: простая прогонка и продольно-поперечная прогонка.

6.1. В первой задаче вычисление значений функций в узловых точках (i , j) для итерации S ведется по формулам вида:

$$\Phi_{ij}^{s} = f(\Phi_{ij}^{s-1}, \Phi_{i-1,j}^{s-1}, \Phi_{i,j-1}^{s-1}, \Phi_{i+1,j}^{s}, \Phi_{i,j+1}^{s}) . \qquad (4.22)$$

В этом случае для прямоугольной области с $m \cdot n$ узловыми точками и тем же разбиением между ℓ ветвями, что и в п.5, вычисление ведется следующим образом.

На первом шаге в первой ветви вычисляется τ точек. Все остальные ветви выполняют пустые операции. После этого происходит передача последнего вычисленного значения функции из первой ветви во вторую. На втором шаге счет идет в двух первых ветвях, на третьем – в трех и, наконец, на ℓ – м – работают все ветви.

Можно видеть, что коэффициент эффективности при этом будет равен

$$\delta = \frac{1}{1+\ell/mS} , \qquad (4.23)$$

где S – общее число итераций.
Как правило, $\ell << mS$, поэтому $\delta \approx 1$, т.е. данная схема алгоритма может считаться эффективной.

6.2. При продольно-поперечной прогонке для той же области и при том же разделении точек между ветвями, что и ранее, вычисления ведутся по формулам вида:

$$\Phi_{ij}^{s} = f_1(\Phi_{ij}^{s-1}, \Phi_{i,j-1}^{s}, \Phi_{i,j+1}^{s-1}) ,$$
$$\Phi_{ij}^{'} = f_2(\Phi_{ij}^{s-1}, \Phi_{i,j-1}^{s-1}, \Phi_{i,j+1}^{s}) , \qquad (4.24)$$

$$\Phi^{s}_{i,j} = f_3(\Phi^{s-1}_{i,j}, \Phi^{s}_{i-1,j}, \Phi^{s-1}_{i+1,j}),$$
$$\Phi^{4}_{i,j} = f_4(\Phi^{s-1}_{i,j}, \Phi^{s-1}_{i-1,j}, \Phi^{s}_{i+1,j}). \qquad (4.25)$$

При счете по формулам (4.24) (продольная прогонка) все ветви ведут вычисления независимо друг от друга.

Счет по формулам (4.25) (поперечная прогонка) удобно вести по той же схеме, что и для рассмотренной выше простой прогонки, т.е. и в этом случае схема р-алгоритма оказывается эффективной.

7. Решение информационно-логических (и.-л.) задач. Этот класс задач приобретает все большее значение в связи с автоматизацией многих областей трудовой деятельности человека, где возникает необходимость иметь дело с большими информационными массивами, их взаимодействием и преобразованием.

7.I. Один из характерных видов взаимодействия массивов, присутствующий в той или иной форме во всех и.-л. задачах большого объема, - перекрещивание массивов.

Под перекрещиванием массивов M_1 и M_2 понимается процесс, при котором каждая пара кодов $a_i \in M_1$ и $b_j \in M_2$ хотя бы однажды одновременно находится в оперативной памяти ЭВМ.

Автором[*)] в свое время было установлено, что время решения данной задачи на ЭВМ будет наименьшим, если массив, вводимый во внутреннем цикле, вводится в оперативную память в γ_{opt} раз меньшими частями, чем массив, вводимый во внешнем цикле, где

$$\gamma_{opt} = \sqrt{S\frac{t_1}{\tau_1} + 1 + \frac{\tau_2}{\tau_1}\frac{S}{m}} \,. \qquad (4.26)$$

---

*) Ю.Г. Косарев. Решение информационно-логических задач. Отчет ИМ СО АН СССР, 1962; Э.В.Евреинов, Ю.Г. Косарев, В.А.Устинов, Применение ЭВМ в исследовании письменности древних майя", т.4, глава 2, Новосибирск, "Наука", 1969.

Здесь S – объем оперативной памяти, отведенной под части данных массивов, m_1 – объем массива M_1, вводимого во внутреннем цикле, t_1 – время ввода одного кода массива M_1; τ_1 и τ_2 – средние времена ожидания при вводе частей массивов M_1 и M_2, соответственно.

При этом время перекрещивания массивов M_1 и M_2 равно

$$T_{min} = \frac{m_1 m_2}{S^2} (1 + \gamma_{opt})^2 \tau_1 + m_2 t_2 , \qquad (4.27)$$

где m_2 – объем массива M_2, t_2 – время ввода одного кода массива M_2.

Обычно без большой погрешности можно пользоваться более простой формулой

$$T_{min} = \frac{m_1 m_2 t_1}{S} , \qquad (4.28)$$

из которой видно, что время решения задачи обратно пропорционально объему оперативной памяти S.

7.2. Схема параллельного алгоритма для перекрещивания массивов исходит из равномерного распределения массивов M_1 и M_2 между ЭМ

$$\dot{L}_2^{i} \; \dot{L}_1^{ji} \; \dot{A}^{kji} \; \dot{O}_{\ell}^{kji} \; \dot{P}_3(k) \; \dot{P}_1(j) \; \dot{P}_2(i) \; \dot{Я} , \qquad (4.29)$$

где функции р-операторов следующие:

$\dot{L}_1^{ji}$ и $\dot{L}_2^{i}$ вводят в оперативную память ЭМ части массивов M_1 и M_2, соответственно;

$\dot{A}^{kji}$ выполняет некоторую специфичную для каждой конкретной задачи процедуру над элементами массивов M_1 и M_2;

$\dot{O}^{kji}$ реализует парную смежную схему обмена, для обмена между ЭМ частями массива M;

$\dot{P}_1(j)$, $\dot{P}_2(i)$, $\dot{P}_3(k)$ организуют соответствующие циклы.

Можно видеть, что и в этом случае время будет минимальным при выполнении в каждой из ЭМ соотношения (4.26).

По аналогии с (4.28) это время может определяться с некоторым приближением выражением

$$T_{min} \approx \frac{m_1 m_2 t_{вп}}{S\ell^2} + \frac{m_1 m_2 t_{эм}}{S\ell} \;, \qquad (4.30)$$

где $t_{вп}$ и $t_{эм}$ – времена ввода одного кода из вспомогательной памяти данной ЭМ и из оперативной памяти другой ЭМ. Здесь первый член учитывает затраты на ввод массива M_1 из внешней памяти, а второй – на обмены частями массива M_1 между ЭМ.

Обычно $t_{эм} << t_{вп}$, поэтому для малых ℓ второй член будет значительно меньше первого и им можно пренебречь. По мере увеличения ℓ скорость роста выигрыша во времени решения на УВС по сравнению с решением на том же числе изолированных ЭМ будет уменьшаться и выигрыш будет стремиться к $t_{вп} / t_{эм}$. Величина этого отношения может служить одним из критериев для выбора $t_{эм}$ УВС, ориентированных на и.-л. задачи.

7.3. Перекрещивание массивов может быть положено в основу решения многих и.-л. задач.

7.3.1. Составление ведомостей. Задача заключается в том, что нужно из некоторого информационного массива M_1 выбрать необходимые данные и поместить их в определенные места массива M_2 (ведомостей) либо непосредственно, либо после некоторой обработки, например, перекодировки или суммирования с прежним содержимым массива M_2 и т.п.

7.3.2. Поиск информации. В этом случае массивом M_1 служит информационный массив (картотека, словарь, таблица и т.п.), а массивом M_2 – множество запросов.

7.3.3. Упорядочение. При первом вводе массива M_1 устанавливается распределение его элементов по признакам, которые используются для упорядочения, и формируется массив M_2 -мест элементов в упорядоченном массиве. После этого задача становится аналогичной составлению ведомостей.

Данный метод упорядочения оказывается эффективнее широко распространенного метода слияния, пока величина массива M_1 не превышает некоторой величины $m_{кр} = f(t_{вп} / t_{эм})$. Для $m > m_{кр}$ можно путем предварительной сортировки разделить массив M_1 на n массивов объемом $m_i < m_{кр} (i = \overline{1,n})$. После этого каждый из массивов упорядочивается на своей подсистеме. Расчеты показывают, что и в этом случае, как правило, время упорядочения оказывается меньше, чем при использовании алгоритмов слияния.

Отметим также то немаловажное обстоятельство, что подобные алгоритмы упорядочения (по сравнению с алгоритмами слияния), кроме времени счета, заметно снижают износ вспомогательных памятей (особенно ленточного типа) и уменьшают вероятность ошибок, благодаря отсутствию многократных перезаписей массива.

8. Аналогичным образом были рассмотрены и остальные задачи, указанные в табл. 4.1. В итоге данного исследования было установлено следующее:

Все изученные классы задач параллельно реализуемы.

Для всех задач с помощью методики распараллеливания по циклам удалось построить параллельные алгоритмы, обладающие пренебрежимо малыми потерями времени выполнения системных взаимодействий, простотой реализации и, что особенно важно, возможностью построения универсальных программ, настраивающихся на число ЭМ как на параметр.

Оказалось, что многие методы и алгоритмы, первоначально разработанные для ЭВМ, обладают необходимыми свойствами для построения на их основе параллельных алгоритмов.

Было также установлено, что из всего многообразия схем обмена в исследованных классах задач используются всего пять типов (табл. 4.2.): трансляционная (Т - "одна - всем"), трансляционная циклическая (ТЦ - "каждая-всем"), парная смежная (ПС- - "каждая-соседней"), коллекторная (К - "все-одной") и парная нестационарная (ПН - "любая-любой"). При этом подавляющее число задач приходится на первые три. Коллекторная схема употребляется обычно для объединения результатов в конце счета, поэтому она не предъявляет высоких требований к скорости реализации. Потребность в парных нестационарных схемах может возникать, когда нужен срочный доступ к информационным массивам, распределенным между различными ЭМ. Интенсивное использование этих схем может приводить к появлению очередей и простоям ЭМ, поэтому желательно принимать меры для сведения их употребления к минимуму. Одной из таких мер может служить сведение данной задачи к задаче перекрещивания массивов (путем накопления запросов в задачах поиска информации, предварительной статистики в задачах упорядочения и т.п.). Интенсивность ПН схем может быть

Т а б л и ц а 4.1

№№ п/п	Задачи, эффективно решаемые на УВС
1.	Решение систем линейных уравнений: методом простой итерации.
2.	-"- -"- -"- -"- -"- Жордана (2в)
3.	-"- -"- -"- -"- -"- Зейделя
4.	-"- -"- -"- -"- -"- модиф. Зейделя
5.	-"- -"- -"- -"- -"- Самарского
6.	Обращение матриц методом пополнения.
7.	Отыскание коэффициентов характеристического полинома.
8.	Умножение матриц.
9.	Решение систем нелинейных уравнений.
10.	Общая задача линейного программирования.
11.	Транспортная задача.
12.	Численное интегрирование (2в.).
13.	Численное дифференцирование.
14.	Задачи интерполирования.
15.	Решение систем обыкновенных дифф. ур. методом Рунге-Кутта.
16.	Расчет подвесок.
17.	Решение граничных задач для линейных систем.
18.	Решение граничных задач для нелинейных систем.
19.	Задачи Дирихле для эллиптических уравнений.
20.	Краевая задача для параболических уравнений.
21.	Задачи Коши для гиперболических уравнений.
22.	Система дифф. ур. в частных производных (прогноз погоды).
23.	Обтекание тела потоком жидкости.
24.	Моделирование работы многогруппового реактора.
25.	Моделирование термодинамич. процессов газовых АЭС циклов.
26.	Случайный поиск с адаптацией.
27.	Поиск информации в больших массивах.
28.	Составление ведомостей.
29.	"Перекрещивание" массивов (2в.).
30.	Задачи теории статистических решений.
31.	Моделирование неупругого рассеяния электронов.
32.	Моделирование рассеяния энергии ионизирующего излучения.
33.	Обработка сейсмических сигналов.
34.	Задача таксономии.
35.	Упорядочивание массивов.

Т а б л и ц а 4.2.

№№ пп	Классификация задач по типам схем обмена					
	Номера задач	Типы схем обмена				
		Т	ТЦ	ПС	К	ПН
I	4, I0, 27б, 28б, 29б	5	0	0	0	0
2	I, 2а, 3, 6, 9, I0, I5, I6, I8, 26, 30, 3Iа, 34, 35	0	I4	0	0	0
3	8, I3, I9, 2I, 24, 27а, 28б, 2Iа, 32	0	0	9	0	0
4	I2, 3Iб	0	0	0	2	0
5	25	0	0	0	0	I
6	I4, I7, 20, 22, 23, 33	0	6	6	0	0
7	2б, 7	2	0	2	0	0
8	5	I	I	I	0	0
	Итого	8	2I	I8	2	I

Т а б л и ц а 4.3.

Примеры задач, решенных на системе "Минск-222"						
Номера задач	Число команд в программе		Время решения (мин) число машин l			$\frac{t_1 \cdot l}{t_l}$
	общее	дополн.	I	2	3	
I	272	I6	5	I,5	-	I,0 - I,7
2	600	35	-	-	-	I,5 - 2,0
6	656	5I	42	2I,8	5,7	I,0 - 2,5
7	272	I04	-	36	-	I,7 - 2,0
I6	857	45	67	I9	I4	I,6 - I,8
23	800	29	65	25	I6,7	I,4 - I,7
24	II67	50	I95	34	-	2,9
25	I000	I6	I0-23	0,25-0,3	-	20 - 40
3I	400	I0	28 час	I4 час	-	I,0
32	890	8	-	5-I0час	-	

заметно уменьшена с помощью рационального размещения массивов и других приемов, развиваемых в теории массового обслуживания.

Для части задач были составлены программы для УВС "Минск-222" и с помощью специальной методики [9] были определены затраты времени на системные взаимодействия (табл. 4.3).

При этом оказалось, что сложность программирования для УВС - того же порядка, что и для ЭВМ. Число команд системы составляет, как правило, менее 10% от общего числа команд.

Таким образом, УВС оказываются эффективнее ЭВМ последовательного типа при решении многих сложных задач. При этом по мере увеличения объема вычислений преимущества УВС становятся все более очевидными, и, наконец, начиная с некоторых значений параметров задач, ЭВМ последовательного типа вообще оказываются практически не пригодными.

ГЛАВА 5. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ УВС

1. Для начального периода развития вычислительной техники характерно применение ЭВМ последовательного типа с простой архитектурой. Повышение производительности машин данного типа достигалось главным образом путем увеличения быстродействия элементов. Одно из основных достоинств этих ЭВМ - простота их технической и математической эксплуатации, а также, что весьма важно, - простота их математического обеспечения (МО), основное назначение которого заключалось в облегчении подготовки задач к счету (это-разработка алгоритмических языков и соответствующих трансляторов, создание библиотек программ и т.п.).

2. В связи с тем, что увеличение быстродействия элементов стало встречать все возрастающие трудности, начали широко применяться способы повышения производительности, основанные на увеличении коэффициента загрузки каждого из устройств ЭВМ. Наибольшее распространение получили: одновременная обработка пакетов из нескольких задач (или разнородных частей одной большой задачи) и режим коллективного пользования на основе разделения времени. Архитектура таких мультипрограммных ЭВМ стала заметно усложняться, что немедленно сказалось на МО. В частности, появилась необходимость в создании специальных операционных систем для согласования между собой одновременно выполняемых различных независимых процессов. Дальнейшее развитие этого направления наталкивается на большие трудности, связанные как с созданием технических средств, так и с разработкой МО, которое значительно усложнилось и резко возросло по объему.

3. За последние годы все отчетливее проявляется тенденция к возврату к ЭВМ с простой архитектурой. Это минимашины, рассчитанные на решение несложных задач массовыми пользователями, и УВС, состоящие из большого числа ЭВМ с простой архитектурой. Основное назначение УВС - решение сложных задач, но они могут также применяться и для обработки потока задач малого и среднего объемов в режимах пакетной обработки или коллективного пользования.

3.1. МО для УВС состоит из МО ЭВМ, взятой в качестве вычислительного модуля, и системного МО, организующего взаимодействие вычислительных модулей.

МО вычислительного модуля по сложности примерно такое же, как и у ЭВМ последовательного типа. В системном МО, так же как в МО ЭВМ, можно выделить две части: операционную, управляющую процессом прохождения задач, и часть, связанную с подготовкой задач к счету на УВС.

3.2. При реализации на УВС режимов пакетной обработки потока задач и коллективного пользования имеет смысл для каждой задачи полностью предоставлять одну или несколько машин, и в каждой из них одновременно выполнять только один процесс.

Естественно, что при этом функции операционной системы сводятся в основном к распределению машин между задачами (или пользователями) и к пересылкам исходной и результирующей информации. Задача распределения для УВС решается гораздо проще, чем для мультипрограммных ЭВМ, благодаря однородности вычислительных модулей.

Подготовка задач к счету в этих режимах неспецифична, и для этой цели может быть использовано МО вычислительного модуля.

3.3. В режиме решения на УВС одной сложной задачи основную долю системного МО составляет подготовка задач к счету. Функции операционной системы сведены к минимуму, и она гораздо проще, чем в двух предыдущих режимах. Это важное свойство обуславливается тем, что для сложных задач (как показано в гл. 3 и 4) характерна многоуровневая циклическая структура, допускающая е с т е с т в е н н о е разделение процесса решения на однородные и относительно автономные ветви вычислений, хорошо согласующееся с функциональной структурой УВС. При этом взаимодействия между ветвями вычислений сводятся, как показано в гл. 4, к нескольким типовым процедурам. Включение этих процедур в параллельный алгоритм решения задачи избавляет от необходимости вводить сложные специальные средства МО для управления процессом решения, как это приходится делать в мультипрограммных ЭВМ. Вместе с тем это позволяет при разработке системного МО широко использовать привычное МО для ЭВМ.

4. В диссертации приведены примеры входных языков для УВС. Эти языки представляют собой расширение известных алгоритмических языков за счет включения в них средств для описания взаимодействий между параллельными ветвями вычислений, которое, как уже упоминалось, может быть сведено к небольшому числу операторов,

реализуемых с помощью сравнительно простых стандартных процедур (табл. 5.I).

Благодаря этому, описание входных программ для УВС мало отличается от привычных описаний для ЭВМ.

5. Трансляторы для УВС также имеет смысл строить из двух блоков: системного и модульного. Эти блоки могут работать независимо друг от друга, благодаря чему в качестве модульного блока могут служить без каких-либо изменений обычные трансляторы для ЭВМ, взятой в качестве вычислительного модуля. Опыт программирования для УВС "Минск-222" с помощью построенных таким образом трансляторов с языков типа АЛГОЛ, автокод АКИ и ЛЯПАС [I3] показал, что системный блок обеспечивает достаточно высокое качество реализации системных взаимодействий. Затраты времени на системные взаимодействия, так же как и при ручном программировании, оказались пренебрежимо малыми, а соответствующий им объем записи во входной программе обычно не превышает 20% от общей длины входной программы.

Таким образом, качество параллельных программ почти полностью определяется модульным блоком. Известно, что программы, составленные транслятором, как правило, значительно уступают по качеству составленным высококвалифицированными программистами. Поэтому для УВС (так же как и для ЭВМ) при решении задач, критичных по времени счета, пока целесообразно применять программирование в машинных кодах (или языках типа систем символического кодирования) по крайней мере для тех ее частей, на которые приходится основное время счета.

6. Для УВС, так же как и для ЭВМ, большое значение для экономии времени на программирование и улучшения его качества имеет применение библиотек программ.

В эту библиотеку имеет смысл включить стандартные программы системных взаимодействий, служебные системные программы (отладочные, анализирующие и т.п.), универсальные параллельные программы для решения основных задач (например, для решения систем алгебраических уравнений и т.п.) и типовые программы для решения сложных многовариантных практических задач.

7. При разработке МО для УВС в значительной степени было использовано МО для ЭВМ - вычислительного модуля. Такой подход позволил существенно упростить задачу. Основу большинства бло-

Т а б л и ц а 5.I

Пример стандартных системных процедур
для УВС "Минск-222"

Название	№ СП	Формальные параметры	Параметры
Настройка	50	("40", AI, A2)	AI – содержимое I-го адреса команды настройки; A2 – содержимое 2-го адреса команды настройки
Обмен	5I	("4I",T,K,E, j)	T = -46, если машина передающая; T = -47,если машина принимающая; K-число передаваемых (принимаемых) кодов; E-передаваемый(принимаемый) массив; j=(i -I) – номер i -го элемента, начиная с которого происходит передача (прием) (i =I,2 , 3,...)
ОБП	52	("42",X,У,A,AI,A2)	X = 32 при ОБП (I); X = 0 при ОБП (0); У – код команды, передаваемый по ОБП. Равен любой операции, кроме операции с ПФЛ и БП; A = 0, если код со знаком (+); A = -0, если код со знаком (-); AI и A2 – соответственно первый и второй адреса передаваемой команды.
	53	("43",X,У,AI,A2)	Применяется для передачи по ОБП операции с ПФЛ и БП. Назначение параметров X,У,A2 аналогичное с СП-52; AI-условное число в командах -60, -61; AI=0 – в остальных случаях.
$ОУП_0$	54	("44", "I")	Синхронизация. Число "I" связано с особенностями транслятора.
$ОУП_i$ (i = I,2,3)	55	("45",P, $L2$)	P:= -1 в случае переполнения,отрицательного знака, нуля. P:= 1 в противном случае. $L2$ – адрес передачи управления при Ω = I.

ков типового состава МО для УВС (рис. 5.I) составляют при этом аналогичные блоки МО для ЭВМ. Благодаря этому, как показало моделирование многих из этих блоков на УВС "Минск-222", объем МО УВС отличается от объема МО ЭВМ не более чем на 10% (табл. 5.2).

8. При разработке МО для УВС особое внимание было уделено универсальным типовым программам, на выполнение которых приходятся основные затраты времени работы УВС.

Программы этого типа используются при моделировании сложных физических процессов, при проектировании сложных физико-технических объектов, при распознавании образов, при решении сложных информационно-логических задач и т.д.

Особенность этих задач-многовариантность и высокие требования к скорости решения. Как правило,для этих задач характерна сравнительно невысокая точность исходной и результирующей информации. Задачи этого класса обладают обычно многоуровневой циклической структурой и содержат во внутреннем цикле сложные функции, счет которых составляет основное время решения задачи.

Для решения этого класса задач используются методы структурного моделирования. Применительно к УВС сущность этих методов заключается в выборе из множества структурных моделей решения данной задачи модели, наилучшим образом отображающейся функциональной структурой УВС. При этом учитывается как коэффициент загрузки вычислителей δ, так и коэффициент использования результатов предыдущих вычислений μ, характеризующий качество распределения памятей:

$$\mu = \frac{\mathcal{X}}{\mathcal{X} - R}, \qquad (5.I)$$

где $\mathcal{X}$ – общий объем вычислений, R – среднее число операций, которое выигрывается в результате замены счета запоминанием.

При этом могут быть два основных случая : запоминание результатов вычислений, которые в дальнейшем многократно используются, и применение таблиц функций, насчитываемых заранее.

9. В диссертации исследованы особенности использования методов структурного моделирования для ряда типовых задач,решенных на системе "Минск-222".

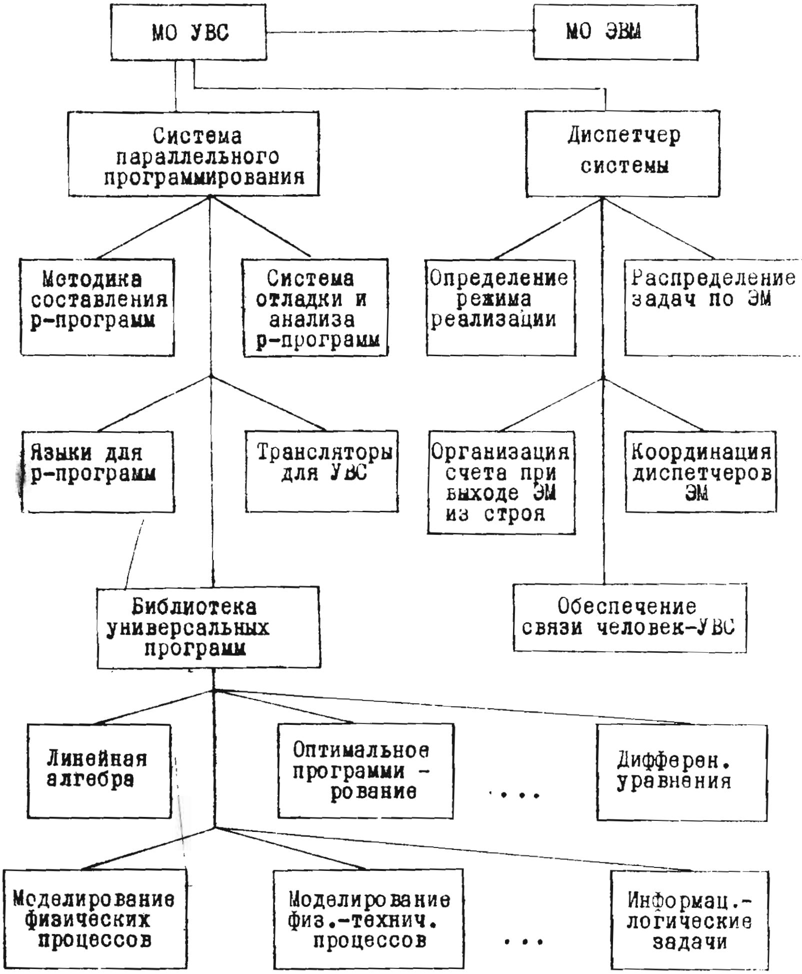
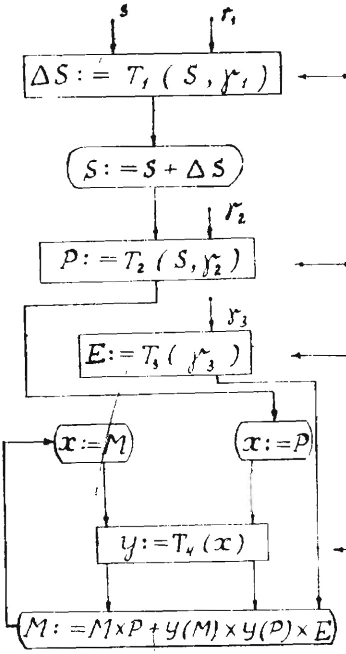


Рис.5.I. Состав математического обеспечения для УВС.

Таблица 5.2

№№ пп	Компоненты математического обеспечения УВС	Общий объем	Объем системной части
I.	Основные диспетчерские блоки системы	2000	2000
2.	Транслятор с АЛГОЛа (ТАМ - 2/22)	I6000	300 - 500
3.	Транслятор с ЛЯПАСа	4000	300 - 500
4.	Транслятор с АКИ	4000	300 - 500
5.	Программа отладки и анализа системных программ	2500	I000
6.	Универсальные р-программы для решения задач линейной алгебры и линейного программирования	3000	250
7.	Универсальные р-программы для решения систем дифференциальных уравнений	2200	80
8.	Универсальные р-программы для решения информационно-логических задач	3500	I60
9.	Универсальные типовые р-программы для моделирования сложных процессов	4000	75
	Итого	4I500	4500 - 5000

9.1. Моделирование неупругого рассеяния электронов методом Монте-Карло на тонком слое вещества [16]. Схематически модель розыгрыша одного из актов взаимодействия электрона с атомами вещества можно изобразить так:



$$\int_S^{S+\Delta S} \Sigma_1(s)\,ds = -\ln \gamma_1 \qquad (5.1)$$

$$\int_{-1}^{P} \Sigma_2(S,P)\,dP = \gamma_2 \qquad (5.2)$$

$$E = \cos 2\pi \gamma_3 \qquad (5.3)$$

$$y = \sqrt{1-x^2} \qquad (5.4)$$

Здесь $T_1, \ldots, T_4$ - табличные функции; $\Sigma_1(S)$ и $\Sigma_2(S,P)$ - функции, зависящие от параметров вещества и начальной энергии электрона; ΔS - длина свободного пробега, $M = \cos\theta$, $E = \cos\varphi$, где θ - угол между отрезком траектории электрона и нормалью к слою, φ - азимутальный угол; γ_1, γ_2, γ_3 - случайные числа, равномерно распределенные в интервале (0,1).

В данной структурной модели выделен основной цикл, включающий в себя счет функций (5.2) - (5.5). Непосредственное вычисление этих функций потребовало бы при счете для каждого варианта задания начальных параметров $\approx 10^4$ операций. При моделировании приходится считать 10^6-10^7 актов соударения.

Как показало исследование, требуемая точность вычисления этих функций позволяет представить их в табличном виде. Таблицы T_3 и T_4 - постоянные, имеют простую ассоциативную выборку, объем каждой из них 1024 ячейки.

Таблицы T_I и T_2 - от двух аргументов, объемом 64000 - - 128000 каждая. Выборка из таблиц как ассоциативная, так и адресная. В среднем выполнение операции выборки требует 2-3 команд.

Таблицы распределялись по вычислительным модулям, каждый из которых вел счет своей части испытаний. Применение таблиц ($\mu \approx 10^2$) и эффективное распараллеливание ($\delta \approx 1$) сделало реальным решение данной задачи на УВС "Минск-222".

9.2. Моделирование термодинамических циклов диссоциирующих газов при расчете процессов теплообмена атомных электростанций [19].

В этой задаче, решенной совместно с сотрудниками ИЯЭ АН БССР, основное время затрачивалось на определение энтальпии $\mathcal{J}$ и энтропии S. Вычисление велось по следующим формулам:

$$\ln K_{p10} = f_1(T), \quad \ln K_{p20} = f_2(T);$$

$$\gamma_j = F(P/P_j, T/T_j); \tag{5.6}$$

$$K_{p1} = K_{p10}/\gamma_1, \quad K_{p20}\gamma_1^2/(\gamma_3^2 \cdot \gamma_4)$$

$$\alpha_1 = (1 \ \alpha_2)^3 K_{p2}/[\alpha_2^3 P - K_{p2}(1+\alpha_2)(1-\alpha_2)^2],$$

где $0 \le \alpha_2 \le 1$ - корень полинома

$$\sum_{i=0}^{6} b_i(K_{p1}, K_{p2}, P)\alpha_2^i = 0;$$

$$i_j = f_{3j}(T), \quad s_j = f_{4j}(T);$$

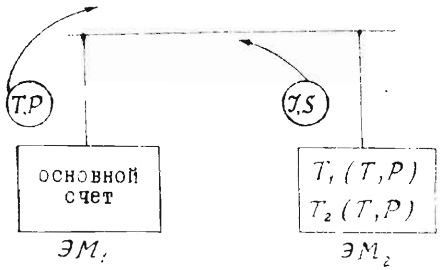
$$\Delta H_j = \frac{RT^2}{\gamma_j} \ \frac{d\gamma_j}{dT}, \quad \Delta S = \frac{\Delta H_j}{T} + R\ln\gamma_j;$$

$$\mathcal{J}_j = i_j - \Delta H_j + H_j, \quad S_j = S_j - \Delta S_j + S_j^o;$$

$$\mathcal{J} \quad f_5(\mathcal{J}_j, \alpha_1, \alpha_2), \quad S \ \ f_6(S_j, \alpha_1, \alpha_2).$$

Здесь R , H° , S_i° - константы, f_1 ,...., f_6 - рациональные функции от аргументов и их логарифмов, F - табличная функция , j = I,2,3,4.

Для сокращения времени счета по формулам (5.6) были составлены таблицы J $T_1(T,P)$ и $S = T_2(T,P)$. Эти таблицы заняли почти всю память одной из ЭМ системы "Минск-222", которая работала в режиме справочника



Несмотря на то, что одна из ЭМ всегда простаивала (δ=0,5), объединение ЭВМ в систему дало общий выигрыш во времени по сравнению с изолированными ЭВМ благодаря применению таблиц (μ =40-76) в 20-38 раз, что позволило сэкономить несколько тысяч часов машинного времени.

9.3. Моделирование колебательных процессов подвесок автопоездов. Задача выполнялась для Минского автозавода совместно с сотрудником ИМ АН БССР А.И. Петрович [I4] . Данная задача сводится к решению системы дифференциальных уравнений 2-го порядка, правые части которых содержат табличные функции, характеризующие микропрофиль дороги. Основное время счета на одной ЭВМ занимали выборка функций из таблиц, которые имели неравномерный шаг, и их интерполирование.

Применение системы "Минск-222" позволило использовать таблицы с достаточно малым и равномерным шагом, что исключило необходимость в интерполировании и свело выборку из таблицы всего к двум командам. Уравнения вместе с соответствующими таблицами равномерно распределялись между машинами ($\sigma \approx 0,9$). При счете на системе из двух ЭМ благодаря использованию таблиц с ассоциативной выборкой ($\mu \approx 2$) получается выигрыш во времени в 3,9 раза по сравнению с одной ЭВМ "Минск-22".

ЗАКЛЮЧЕНИЕ

В диссертации решены три тесно связанные проблемы, возникающие при создании высокопроизводительных вычислительных систем на основе принципов параллельности, однородности и программной изменяемости структуры:

- синтез функциональных структур технических средств,
- синтез структурных моделей для решения сложных задач,
- отображение структурных моделей процесса решения задач на функциональную структуру вычислительной системы (системное математическое обеспечение).

По всем трем проблемам путем сочетания теоретических и экспериментальных исследований удалось преодолеть основные препятствия, стоящие на пути разработки высокопроизводительных универсальных однородных вычислительных систем с управляемыми связями, что открывает возможность их построения уже в ближайшие годы.

В результате выполнения данной работы были получены следующие основные результаты:

1. Предложены простые в изготовлении и использовании магистральные вычислительные системы, эффективно реализующие параллельные алгоритмы решения сложных задач и обладающие свойством сохранять работоспособность при изменении в широком диапазоне количества вычислительных модулей. Эти системы могут найти применение как для построения средств вычислений высокой производительности, так и для создания устройств, обладающих высокой структурной надежностью и живучестью.

2. На основе анализа большого числа классов сложных задач установлено, что все параллельные алгоритмы, как правило, используют три схемы обмена информацией между ветвями вы-

числений: трансляционную ("одна - всем"), трансляционную циклическую ("каждая - всем") и парную смежную ("каждая - соседним").

Путем теоретико-вероятностных расчетов и экспериментальных исследований(с помощью специальных программ)показано, что общие затраты на системные взаимодействия составляют небольшую долю от общего времени решения задач.

3. Доказано, что с помощью набора команд системы: „передача", „прием", „синхронизация", „обобщенный условный переход", „обобщенный безусловный переход", „настройка" может быть получен эффективный и в то же время технически простой механизм образования заданных функциональных структур вычислительных систем.

4. В соответствии с теоретическими положениями, развитыми автором, на базе серийной машины "Минск-22" разработана и создана универсальная однородная мономагистральная вычислительная система с управляемыми связями "Минск-222", послужившая основой для создания однородных вычислительных систем в ряде организаций страны.

5. Создана простая и эффективная методика построения универсальных параллельных программ,параметрически настраивающихся на число вычислительных модулей в УВС.

- Изучены свойства параллельных алгоритмов.

- Разработан язык и его модификации для описания схем параллельных алгоритмов.

- предложен набор стандартных процедур для реализации системных взаимодействий. Добавление этих процедур к алгоритмическим входным языкам (АЛГОЛу, автокоду, ЛЯПАСу и др.) позволило создать простой способ представления входных параллельных программ.

6. Исследованы, разработаны и экспериментально проверены на системе "Минск-222" принципы организации и состав системного математического обеспечения.

На базе имеющихся трансляторов для трех типов языков: АЛГОЛ, автокод АКИ, ЛЯПАС-разработаны трансляторы для системы.

- Разработаны блоки системного диспетчера, обеспечивающие работу системы в режиме пакетной обработки потока задач средней и малой сложности и в режиме коллективного пользования.

- Разработаны программы отладки и анализа параллельных программ и т.д.

7. Построены, исследованы и классифицированы структурные модели большого числа классов сложных задач.

На этой основе выявлены требования к техническим средствам и системному математическому обеспечению.

Показано, что основные классы задач эффективно реализуются на вычислительных системах с магистральными связями с помощью методики распараллеливания по циклам.

Установлено, что все многообразие структурных моделей процесса решения задач сводится к трем основным типам: однородному, конвейерному и смешанному.

8. Разработана и экспериментально проверена на системе "Минск-222" методика решения сложных задач, сочетающая одновременную работу процессоров с эффективным использованием суммарного объема памяти системы.

Показано, что время решения задач на машинах, объединенных в систему, как правило, значительно меньше времени решения на том же числе одиночных машин.

На основе данной методики разработаны типовые программы для решения ряда важных классов практических задач на системе "Минск-222".

По теме диссертации автором опубликованы две монографии [1,2] и ряд статей [3-25]. Материалы по конкретным разработкам вычислительных систем содержатся в отчетах [26-28].

Основные результаты, изложенные в диссертации, обсуждались на Всесоюзном симпозиуме по вычислительным системам, на I и II Всесоюзных конференциях по вычислительным системам, на Всесоюзной конференции по системотехнике, на Конференциях общества им. А.С.Попова, на семинарах в Институте кибернетике АН УССР и в ряде организаций Министерств Радиотехнической промышленности, электронной промышленности, Обороны и других ведомств.

Цикл работ по разработке системы "Минск-222" и методике решения задач на универсальных однородных вычислительных системах с программно изменяемыми связями получил положительную оценку Междуведомственной комиссии.

Основные работы по теме диссертации

1. Ю.Г.Косарев, Э.В.Евреинов. О возможности построения вычислительных систем высокой производительности. Новосибирск, Изд-во СО АН СССР, 1962.
2. Ю.Г.Косарев, Э.В.Евреинов.Однородные универсальные вычислительные системы высокой производительности. Новосибирск, "Наука", 1966.
3. Ю.Г.Косарев, Э.В.Евреинов. О системах автоматизации научных экспериментов для разработки вычислительных систем.-Вычислительные системы, Новосибирск, 1963, вып. 8,3-10.
4. Ю.Г.Косарев, Э.В.Евреинов.О вычислительных системах высокой производительности. - Изд-во АН СССР, сер.тех.кибернетика, 1963, № 4, стр. 3-25.
5. Ю.Г.Косарев, Э.В.Евреинов.Матричный р-язык для описания параллельных алгоритмов. - Вычислительные системы, Новосибирск, 1965, вып. 17, стр. 100-105.
6. Ю.Г.Косарев, Э.В.Евреинов.О решении задач на универсальных вычислительных системах. - Вычислительные системы, Новосибирск, "Наука", 1965, вып. 17, стр. 106-164.
7. Ю.Г.Косарев. О методике решения задач на универсальных вычислительных системах. - Вычислительные системы,Новосибирск, "Наука", 1965, вып. 17, стр. 61-99.
8. Ю.Г.Косарев, Э.В.Евреинов. О методике разработки вычислительных систем. -Вычислительные системы, Новосибирск, "Наука", 1963, вып. 6, стр. 3-20.
9. Ю.Г.Косарев. Распараллеливание по циклам. - Вычислительные системы,Новосибирск,"Наука", 1967,вып.24, стр. 3-19.
10. Ю.Г.Косарев, С.В.Нагаев. О потерях времени на синхронизацию в однородных вычислительных системах. -Вычислительные системы, Новосибирск, "Наука", 1967, вып.24, стр. 21-39.

11. Ю.Г. Косарев, В.Л. Пихтин, Е.Н. Жуков , Л.В.Головишкина, Ю.И. Колосова. Об особенности употребления команд системы "Минск-222". - Вычислительные системы, Новосибирск, "Наука", вып. 24, стр. 41-54.
12. Ю.Г. Косарев, Л.В. Головяшкина. Решение системы линейных уравнений на "Минск-222". - Вычислительные системы , Новосибирск, 1967, вып 24, стр. 55-75.
13. Ю.Г. Косарев, Л.В. Головяшкина, Ю.И.Колосова, Н.Н.Миренков. Автоматизация программирования для системы на основе существующих трансляторов. - Вычислительные системы, Новосибирск, "Наука", 1968, вып.30, стр. 63-70.
14. Ю.Г. Косарев, А.А.Петрович.Исследование колебательных процессов автопоездов на системе "Минск-222". - Вычислительные системы, Новосибирск, "Наука", 1968, вып. 30 , стр. 12-15.
15. Ю.Г. Косарев, А.А. Велесько, Р.В. Юревич. Многогрупповой расчет двумерного реактора в диффузионном приближении на системе "Минск-222". - Вычислительные системы,Новосибирск, "Наука", 1968, вып. 30, стр. 15-22.
16. Ю.Г. Косарев.Моделирование неупругого рассеяния электронов методом Монте-Карло. - Вычислительные системы, Новосибирск, "Наука", 1968, вып. 30, стр. 34-41.
17. Ю.Г. Косарев, Е.Л. Ющенко. Некоторые проблемы организации вычислительного процесса на системах. - Труды семинара "Автоматизация программирования", Киев,1968,вып.1, стр. 3-6.
18. Ю.Г. Косарев, Ю.И. Колосова, В.А. Казущик. Измерение временных характеристик программы системы. -Вычислитель - ные системы, Новосибирск, "Наука", 1968, вып. 30, стр. 55-63.
19. Ю.Г. Косарев, М.А. Бажин, В.П. Бубнов, И.С.Захарова. Термодинамический расчет газовых АЭС циклов на диссоциирующих газах на системе "Минск-222". -Вычислительные системы, Новосибирск, "Наука", 1968, вып. 30, стр. 22-26.
20. Ю.Г. Косарев, В.А. Наумов, С.Г. Розин, А.А. Ярошевич. Решение задачи о рассеянии энергии ионизирующего излучения методом Монте-Карло на системе "Минск-222". - Вычислительные системы, Новосибирск, "Наука", 1968, вып. 30, стр. 41-46.

21. Ю.Г. Косарев. Примеры использования таблиц для сокращения времени счета. - Вычислительные системы, Новосибирск, "Наука", 1968, вып. 30, стр. 46-55.
22. Ю.Г. Косарев, В.Н. Гушенсков, Л.В. Жаврид, В.А. Казущик, Н.П. Савик. Усовершенствование системы команд вычислительной системы "Минск-222". - Вычислительные системы, Новосибирск, 1970, вып. 42, стр. 74-80.
23. Ю.Г. Косарев, И.В. Вельбицкий. Системный подход к построению трансляторов для вычислительных систем.- Вычислительные системы, Новосибирск, 1970, вып.42, стр.12-21.
24. Ю.Г. Косарев, Н.В. Кучин. Параллельный алгоритм для решения задачи таксономии. - Вычислительные системы, Новосибирск, 1970, вып. 42, стр. 3-11.
25. Ю.Г. Косарев. О структурах вычислительных систем, устойчивых к изменению числа машин. - Вычислительные системы, Новосибирск, 1970, вып. 42, стр. 59-74.
26. Ю.Г. Косарев, Э.В. Евреинов, В.Г. Хорошевский. Отчет ИМ СО АН СССР, № 1.1.15, 1967, 290 стр.
27. Ю.Г. Косарев, Э.В. Евреинов. Отчет ИМ СО АН СССР, № 117, 1968, 377 стр.
28. Ю.Г. Косарев, Э.В. Евреинов. Отчет ИМ СО АН СССР, № 421, 1969, 151 стр.

Заказ 503. Объем 3 п.л., 2 уч.изд.л.
Тираж 110 экз.

Отпечатано в Институте математики СО АН СССР, Новосибирск,90