

Глава 7. Место и роль алгоритмического базиса в решении проблемы производительности

7.1. Алгоритмический базис и его классификация

В данной главе основное внимание будет уделено исследованию перспектив развития алгоритмического базиса. Более подробно перспективы развития элементарно-технологического базиса приведены в начале главы 10. Перспективы развития организационного базиса – в главах 4, 9 – 11. Перспективы развития информационного базиса приведены в работе [1]. Экономические аспекты проблемы производительности компьютерных средств приведены в работе [2].

С момента появления ЭВМ все более усиливается их влияние на все области деятельности человечества.

Одна из основных закономерностей, определяющих развитие вычислительных средств, – постоянное повышение их производительности. В связи с этим проблема производительности является центральной в компьютерной науке и ее решение тесно связано с совершенствованием элементарно-технологического, организационного, информационного и алгоритмического базисов.

При этом необходимо учитывать слова корабеля и математика академика А.Н. Крылова [11]: «В приложениях обыкновенно интересуют не процесс вычислений, а результат его, поэтому и стараются получить этот результат с достаточной точностью при наименьшей затрате труда и времени».

В виду ограниченности средств совершенствования большинства вышеперечисленных базисов в решении проблемы производительности на современном этапе и в будущем важную роль играет совершенствование алгоритмического базиса. В этой связи справедливо положение [12]: «Просмотр методов решения сложных прикладных задач показывает, что, как правило, эффект, достигаемый за счет совершенствования численных методов, по порядку сравним с эффектом, достигаемым за счет повышения производительности ЭВМ». Аналогичное можно утверждать и при совершенствовании наиболее массовых алгоритмов. Поэтому колоссальный рост скорости вычислений не уменьшает значения эффективных алгоритмов, в которых в большинстве прикладных областей ощущается острая нехватка.

В этой связи имеют важное значение рассмотренные в данной работе вопросы анализа базисов вычислительной техники и классификация алгоритмического базиса по критериям массовости, адаптивности, интенсивности и т.д.

Широкий спектр использования алгоритмов, а также учет особенностей различных видов вычислительных средств, решаемых задач, экономических условий применения и критериев их эффективности сделали актуальным введение понятия алгоритмического базиса.

Целесообразным является рассмотрение периодов развития алгоритмического базиса, связанное с развитием товарного производства информационной продукции.

С точки зрения экономики, подобно материальному производству, можно выделить следующие пять периодов развития алгоритмического базиса:

- создание алгоритмов для удовлетворения собственных потребностей разработчиков технологических и программных средств;
- становление товарного производства алгоритмических и программных средств;
- экстенсивное товарное производство средств вычислительной техники, алгоритмов и программ;
- становление интенсивного товарного производства средств вычислительной техники, алгоритмов и программ;
- интенсивное товарное производство обеспечения средств вычислительной техники, алгоритмов и программ.

В настоящее время реализуется четвертый период развития алгоритмического базиса и средств вычислительной техники.

До тех пор, пока алгоритмы разрабатывались для собственных потребностей пользователя, введение этого понятия не было необходимым. Однако становление товарного производства информационной продукции выдвинуло на передний план технологичность алгоритма – совокупность его свойств, обеспечивающих минимизацию времени счета и затрат ресурсов при производстве информационной продукции. Благодаря технологичности уменьшается себестоимость производимой информационной продукции и увеличивается экономичность и эффективность используемого алгоритма по сравнению с другими вариантами. При этом важным качеством технологичности алгоритма является адаптивность к внешним и внутренним условиям применения. Поэтому вполне естественно, что вопросы решения проблемы производительности на основе алгоритмического базиса в полном объеме не ставились и не решались.

При производстве информационной продукции вообще и при создании вычислительных алгоритмов, в частности, используются следующие факторы:

- разработка, исследование и реализация новых методов и алгоритмов, обеспечивающих повышение точности, скорости сходимости, численной устойчивости и сокращения времени счета задачи;
- обеспечение ресурсосбережения (минимизация допустимой точности счета, экономия памяти и других ресурсов);
- снижение трудоемкости производимой информационной продукции за счет адаптации (односторонней либо взаимной) алгоритмических, информационных и технических вычислительных средств;
- повышение фондоотдачи информационной продукции за счет выявления и устранения «узких мест» и диспропорций в вычислительном процессе,

максимизация загрузки устройств обработки, устранение непроизводительных затрат и простоев оборудования путем использования соответствующей организации процесса вычислений и создания соответствующих алгоритмов, использование динамических схем счета и т.д.;

– аппаратная либо микропрограммная реализация наиболее массовых алгоритмов нижнего уровня или их частичная аппаратная поддержка;

– широкое использование прошлого труда (использование табличных, таблично-алгоритмических методов и т.д.);

– интеллектуализация при создании алгоритмов и их реализация.

В общем случае проблема выбора эффективного алгоритма включает исследование следующих множеств [13]: задач, алгоритмов, свойств и критериев. При этом ускорение вычислений во многом зависит от использования разнообразия алгоритмов решаемых задач, массовости использования и их адаптации к условиям применения. Именно эти три таксона лежат в основе классификации алгоритмического базиса как фактора процесса ускорения вычислений.

В данной работе под алгоритмическим базисом¹ понимается совокупность методов и схем счета, алгоритмов, функциональных преобразований, решающих правил, моделей вычислительных процессов, стандартных и определяемых функций, выражений, цепочек операторов, макросов и т.д., составляющих основу решения прикладных задач.

Классификация алгоритмического базиса может быть осуществлена исходя из различных подходов: функционального, статистического, адаптации к условиям применения, учета особенностей алгоритмов по предметным областям использования, закрепления выполняемых функций в вычислительных средствах, системного, генетического, динамического, статического, микро- и макро подходов, сложности алгоритмов, экономического и др.

Однако для решения проблемы производительности на основе совершенствования алгоритмического базиса целесообразно рассмотреть классификации, основанные на функциональных, статических и адаптивных подходах, с учетом в той или иной мере остальных подходов.

Рассмотрим эти классификации, исходя из различных подходов и методов декомпозиции алгоритмического базиса.

Классификация алгоритмов, исходя из особенностей классов решаемых задач и частоты встречаемости операций (операторов), является для современной компьютерной науки привычной. Например, определение эффективной производительности по смесям Гибсона; деление задач по классам и внутри их и т.д. Выявление таких особенностей чрезвычайно важно при построении специализированных и проблемно-ориентированных вычислительных средств.

¹ В работе под термином «базис» понимается совокупность определенных средств и ресурсов, участвующих в процессе производства информационного продукта.

Классификация алгоритмов по критерию массовости, которая отражает свойство использования алгоритма в задачах определенного класса, также известна компьютерной науке, но не связывалась напрямую с повышением производительности вычислительных средств.

Подобная классификация впервые осуществлена для таких важных классов задач, как обработка сигналов и изображений. Это объясняется необходимостью обработки больших массивов информации и во многих случаях необходимостью решения этих задач в реальном времени. В данной работе критерий массовости использования алгоритмов применяется для всех классов задач и имеет новое значение.

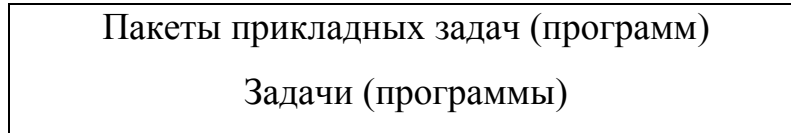
Классификация по критерию адаптивности алгоритма к внешним и внутренним условиям применения направлена на уменьшение временной сложности алгоритма и/или обеспечение универсальности использования. При этом различают адаптацию к обрабатываемым данным, особенностям используемых вычислительных средств и/или классу решаемых задач. Деление задач и методов на классы может быть получено исходя и из разных подходов: по классам гладкости, по используемым данным, по методу получения результата (численные, аналитические, численно-аналитические), на основе количественных характеристик (много входных, мало счета; мало входных данных, много счета и т.д.), на основе учета статистических данных по использованию операций в ЭВМ для различных классов задач (смеси Гибсона) и т.д.

С точки зрения производства информационной продукции наиболее предпочтительным является классификация составляющих частей алгоритмов решения задач на основе свойства массовости. Критерий массовости выбран в связи с тем, что оптимизация массовых алгоритмов в наибольшей степени обеспечивает интенсификацию вычислений. С точки зрения критерия массовости алгоритмический базис можно условно разделить на следующие уровни обработки: нижний (базовый), промежуточный и верхний. Нижний уровень обработки включает наиболее массовые части алгоритма решения класса задач, промежуточный – объединяющие специализированные части алгоритма решения данного класса задач, а верхний – обобщенные части алгоритмов решения класса задач (рис. 7.1).

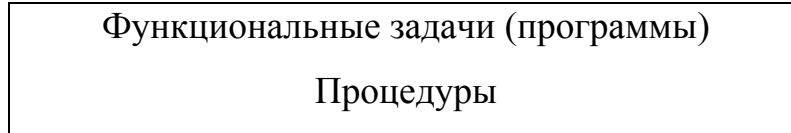
В связи с тем, что наиболее часто используемые фрагменты алгоритмов решения задач относятся к нижнему уровню обработки, то именно они существенно влияют на ускорение вычислений. Это подобно положению Флина о том, что циклы, занимающие менее 4 % кода программы, требуют более половины времени счета задачи. Для сложных задач этот показатель еще выше.

В общем случае в нижний уровень обработки можно включать алгоритмы выполнения операций ЭВМ (при разработке ЭВМ). Но с точки зрения пользователя нижний уровень обработки начинается с операторов.

Верхний уровень обработки:



Промежуточный уровень обработки:



Нижний уровень обработки:

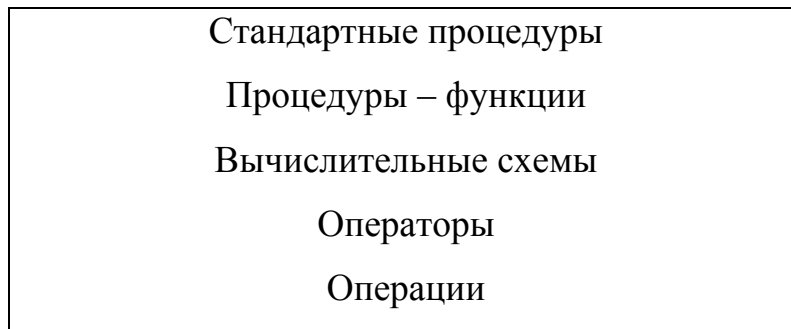


Рис. 7.1. Классификация алгоритмического базиса по критерию массовости использования его компонент

Практика создания языков высокого уровня и математических сопроцессоров показывает, что наиболее часто встречаемые функции либо цепочки операторов включаются в наборы стандартных функций и/или операторы математических сопроцессоров. При решении научно-технических задач в стандартные функции включают, прежде всего, элементарные и некоторые специальные функции, а в некоторых случаях – операторы получения последовательности случайных чисел (для задач моделирования).

В зависимости от рассматриваемой предметной области в различные уровни обработки входят свои подмножества фрагментов алгоритмов решаемых задач.

Существует сильная зависимость производительности системы от соответствия алгоритма решаемой задачи структуре ВС. Поэтому необходимо обеспечить соответствие структуры ВС решаемой задаче (специализация ВС) либо классу задач (проблемная ориентация ВС), либо использовать соответствующие адаптивные методы и алгоритмы. В связи с этим целесообразно рассмотреть классификацию алгоритмов в зависимости от механизмов, позволяющих адаптироваться к условиям применения.

Учитывая эти механизмы приспособления к условиям применения, различают следующие виды адаптации: статическая и динамическая адаптация и автоматический выбор эффективного алгоритма (табл. 7.1). Используя эти механизмы адаптации, можно решить проблему производительности на основе совершенствования алгоритмического базиса.

Таблица 7.1

Классификация алгоритмов по механизмам адаптации

№ п/п	Классы адаптации алгоритмов	Механизмы адаптации
1	2	3
1	Статическая адаптация (предварительная настройка алгоритма)	<p>Выбор наиболее эффективного алгоритма по назначению.</p> <p>Использование прошлого труда (табличные и таблично-алгоритмические методы, вычислительные схемы типа Белаги-Пана и т.д.).</p> <p>Выявление и устранение «узких мест» в вычислительном процессе и т.д.</p>
2	Динамическая адаптация алгоритма к условиям применения	<p>Использование встроенных механизмов адаптации (невязки, итеративные процессы, рекуррентные соотношения, начальные и завершающие приближения).</p> <p>Использование порождающих (обобщенных) алгоритмов.</p> <p>Использование адаптивных по данным алгоритмов (сегментная аппроксимация, динамические схемы счета, методы табулирования и экстраполирования функций и т.д.).</p> <p>Использование адаптивных по точности алгоритмов (переменная либо произвольная разрядность, мини-максные аппроксимации, специальные нормы погрешности, изменяемые константы асимптотики погрешностей и т.д.).</p> <p>Использование адаптивных по времени и ресурсам алгоритмов (различные виды экономичных алго-</p>

Продолжение таблицы 7.1

		ритмов, произвольная сходимость итерационных формул, разнообразные вычислительные схемы и т.д.). Адаптивные по устойчивости счета алгоритмы
3	Автоматический (автоматизированный) выбор эффективного алгоритма для данного применения	Используются механизмы порождения и конкуренции. Используются механизмы обратных связей (обратной и прямой). Используются механизмы саморазвития и самоорганизации. Использование баз знаний нового типа

По способам реализации адаптивные алгоритмы можно реализовать программно (наиболее гибкая, но и наиболее медленная реализация), микропрограммно (достаточно гибкая и быстрая реализация), аппаратно (наименее гибкая, но наиболее быстрая реализация) и комбинируя вышеописанные методы.

Необходимо отметить, что аппаратная реализация адаптивных алгоритмов может быть наиболее эффективной при использовании программируемых вычислительных структур и архитектур.

7.2. Анализ базисов вычислительной техники и решение проблемы производительности

Решение проблемы производительности в общем случае основывается на совершенствовании элементарно-технологического, организационного, информационного и алгоритмического базисов. Средства этих базисов, хотя решают проблему производительности в целом, имеют различную направленность и различные потенциальные возможности. Для оценки роли алгоритмического базиса необходимо ответить на три вопроса.

Первый вопрос, возникающий у исследователей проблемы повышения производительности вычислительных средств:

«Каковы потенциальные возможности совершенствования этих базисов?»

Второй вопрос: «Каковы средства и методы необходимо использовать для совершенствования этих базисов?»

Третий вопрос: «Что дают найденные средства и методы?»

Ответ на первый вопрос важен вследствие того, что средства совершенствования перечисленных выше базисов на каждом из этапов развития вычислительной техники находятся в различных стадиях развития. Ресурс одних

средств может быть близок к истощению, других – неограничен, а третьих – ограничен, но имеет еще достаточно большой резерв развития.

Ответ на второй вопрос дает возможность непосредственно выделить методы и средства, влияющие на повышение производительности вычислительных средств, а ответ на третий вопрос имеет качественную (в рамках выбранного критерия) и количественную оценки.

В целом, ответы на все три вопроса дают возможность оценить место и роль каждого из базисов, методов и средств повышения производительности ЭВМ и ВС.

Как уже отмечалось выше, в вычислительной технике существуют только три вида средств интенсификации вычислений – сокращение количества и продолжительности действий, а также полное или частичное совмещение во времени их выполнения.

В таблице 7.2 приведена направленность методов и средств на интенсификацию процесса вычислений.

Таблица 7.2

Направленность методов и средств Базис	Сокращение общего количества действий	Сокращение продолжительности всех или части действий	Совмещение во времени выполнения действий
1. Элементный	–	+	–
		(всех действий)	
2. Информационный	–	+	+
		(части действий)	
3. Организационный	–	–	+
4. Алгоритмический	+	+	+
		(части действий)	

Из таблицы 7.2 видно, что только алгоритмический базис дает наибольшее число точек приложений. Благодаря этому возрастает роль алгоритмического базиса в интенсификации процесса вычислений.

Проведем анализ достигнутых результатов различными базисами на современном этапе развития компьютерной науки.

Одним из основных показателей, по определению поколений ЭВМ, является степень развития элементного базиса. Непосредственно на производительность вычислительных средств влияют частотные свойства элементного базиса, которые косвенно связаны с миниатюризацией и уменьшением потребляемой мощности.

В настоящее время основой элементной базы вычислительной техники является полупроводниковая электроника. В работе [15] отмечается, что время срабатывания электронных схем на кремниевой основе, применяемых в современных супер – ЭВМ, составляет $5 \cdot 10^{-9} \dots 3,5 \cdot 10^{-10}$ секунды.

Однако при этом главный машинный такт понижается на порядок, что связано с импедансом межсоединений [14]. Повышение быстродействия элементной базы связывают с переходом на арсенид галлия или на использование криогенной техники, что позволит повысить частоту срабатывания логических элементов на 1,5 – 2 порядка [3]. Это утверждение хорошо согласовывается с анализом, проведенным в работе [15], в которой утверждается, что резервы по миниатюризации, быстродействию и уменьшению потребляемых мощностей для элементно-технологической базы составляют 1 – 1,5 порядка на каждый из показателей и близки к исчерпыванию.

Большое значение для создания элементной базы имеет плотность упаковки, уменьшение потребляемой мощности и снижение стоимости.

Так, в начале следующего тысячелетия ожидается создание Гигачипов, которые будут содержать до миллиарда транзисторов [5]. В 2005 году ожидают получения Гигачипов памяти емкостью 4 миллиарда бит (Гигабит), а в 2011 году – 64 Гигабита [5].

Дальнейшее повышение быстродействия элементной базы связывают с созданием приборов на квантово-механических эффектах, где субатомные частицы ведут себя как волны. Эти приборы позволяют повысить упаковку элементов в СБИС на два порядка и на четыре порядка снизить стоимость операции, а также обеспечат уменьшение потребляемой мощности по сравнению с обычными элементами [16].

Большие надежды на повышение быстродействия элементной базы возлагают на создание оптических переключателей с быстродействием 30 ГГц. Оптические устройства, собранные из оптических элементов, могут ускорить считывание информации на 3 порядка за счет параллельного выполнения операций. Началась разработка и исследование действующего образца оптического компьютера, способного обрабатывать данные со скоростью света (университет Хариота-Уатта в Шотландии) [17]. Видимо, исходя из подобных результатов и учета развития вычислительных средств в 80-х годах нашего столетия, в работе [18] утверждается, что освоено лишь 1 – 5 % тех потенциальных возможностей элементной базы, которые можно достичь, приближаясь к фундаментальным физическим пределам. Дальнейшее развитие элементарной базы связывают с нано- и подобными технологиями.

Таким образом, можно утверждать о практической исчерпанности элементного базиса, основанного на полупроводниковой технологии и значительных резервах его развития при использовании оптических переключателей и

нанотехнологий. Однако в целом развитие элементарно-технологического базиса в любом случае ограничено фундаментальными законами физики.

Элементарно-технологический базис определяет только основу создания высокопроизводительной вычислительной техники, но ее эффективное использование во многом зависит от совершенствования остальных базисов.

Перейдем к анализу состояния остальных базисов.

Организационный базис проще всего проанализировать на основе использования коэффициента производительности используемого машинного времени [6]:

$$\eta = T_{\text{п}} / (T_{\text{п}} + T_{\text{н}}),$$

где $T_{\text{п}}, T_{\text{н}}$ – соответственно производительное и непроизводительное машинное время;

$T_{\text{п}} + T_{\text{н}} = T_{\text{общ}}$ – общее время, затрачиваемое вычислительными средствами на решение задач.

Показатель η (аналог КПД) позволяет оценить долю непроизводительных затрат времени на решение задачи. Как было показано в работе [6], для ЭВМ третьего поколения показатель $\eta = 0,01$, т.е. примерно равен 1 %.

Причина таких низких производительных затрат состоит, с одной стороны, в незагруженности операционных автоматов по вине архитектурно-структурных решений, а с другой стороны, в несовершенстве ОС. Так, на выполнение алгоритмов ОС EHEC-8 расходуется 98,3 % производительности ЭВМ UNIVAC 1108 при решении задач реального времени и 88,8 % при решении стандартного набора задач [18]. Однако в отдельных случаях (за счет специализации и адаптации технических алгоритмических средств) удается довести показатель η до 0,8. Теоретически $\max \eta = 1$, т.е. и в этом случае имеется предельная форма организации. Но и при коэффициенте $\eta = 1$ реальная производительность мультипроцессорной системы во многом зависит от алгоритмического базиса и алгоритмов решения задач.

Анализ информационного базиса (совокупность средств представления, хранения и передачи информации) будем проводить в аспекте влияния его на решение проблем производительности.

Скорость передачи информации по линиям связи в пределе ограничена скоростью света, т.е. $3 \cdot 10^5$ км/час = 0,3 м/нс.

Но в реальных линиях связи при длине линии связи 25 – 35 см величина задержки распространения сигнала $t = 2 - 2,5$ нс [14]. В соответствии с законами Амдала, утверждающими, что производительность системы определяется самым медленным ее компонентом, и Флина, что каждому миллиону операций производительности должно соответствовать не менее 1 Мбайта оперативной

памяти, понятно требование «трех Т» (Терафлопс, Терабайт, Терабайт/сек) для будущих супер-ЭВМ [17, 19].

Баланс производительности, емкости памяти и пропускной способности может достигаться различными средствами. Так, время обращения к оперативной памяти определяется, с одной стороны, элементной базой, а с другой стороны, ее архитектурой (иерархическим построением системы памяти ЭВМ и ее расслоением, т.е. организацией ее функционирования). При этом предельное время обращения к памяти также ограничено фундаментальными законами физики. Аналогично можно сказать и о пропускной способности каналов. При этом необходимо учесть, что в реальных системах связи их пропускная способность зависит также от эффективного их использования.

Так, в работе [20] показано, что показатель эффективного использования каналов (аналог КПД для каналов) $\eta = U_n / U_H$, где U_n, U_H – соответственно полезная и номинальная пропускная способность канала. Там же отмечается, что для реальных каналов величина $\eta = 30 - 60\%$. Поэтому для обеспечения роста пропускной способности каналов после исчерпания интенсивных факторов используются экстенсивные факторы – увеличение числа каналов.

Для повышения производительности вычислительных средств важна еще одна компонента информационного базиса – представление информации вообще и в обрабатывающих устройствах, в частности. Для последнего случая подробный анализ проведен в работе [7], где показано, что наибольшим эффектом обладает система проблемно-ориентированных счислений со сверхвысокими основаниями.

Таким образом, и этот вид интенсивных средств повышения производительности вычислительных средств в пределе ограничен и практически исчерпан.

Именно поэтому в основном дальнейшее увеличение производительности ЭВМ связывают с экстенсивными факторами – параллельными вычислениями. Используя современные достижения компьютерной науки и технологии, можно добиться «трех Т» в ближайшее время, а в дальнейшем и более существенных результатов. Но потребности высокопроизводительных и экономических вычислений растут еще быстрее. Поэтому, не умаляя роли параллельных вычислений в решении проблемы производительности, возникает необходимость обращения к нетрадиционному средству интенсификации процесса вычислений – к совершенствованию алгоритмического базиса. При этом необходимо учесть следующие обстоятельства:

- неисчерпаемость совершенствования алгоритмического базиса в силу неисчерпаемости процесса познания;
- направленность совершенствования алгоритмического базиса на интенсификацию процесса вычислений;

- невозможность решения проблемы производительности СВТ без учета возможностей совершенствования алгоритмического базиса;
- возрастающую роль совершенствования алгоритмического базиса по мере исчерпывания возможностей других базисов;
- использование алгоритмического базиса во всех видах вычислительных средств;
- возрастание потенциальных возможностей алгоритмического базиса по мере накопления новых знаний.

7.3. Методы и средства алгоритмического базиса для ускорения процесса вычислений

Как уже отмечалось выше, алгоритмический базис характеризуется выбранными методами решения задач и способами их реализации на ЭВМ и ВС. Алгоритмический базис оказывает существенное влияние на время решения задачи, количество выполняемых устройствами действий (операций), объемы используемой памяти, загрузку каналов и другого оборудования.

Совершенствование алгоритмического базиса для повышения производительности ЭВМ и ВС в настоящее время происходит лишь эпизодически, что в основном объясняется имеющимися пока резервами организационного, элементно-технологического и информационного базисов, а также отсутствием теоретических положений о роли и месте алгоритмического базиса в процессе интенсификации процесса вычислений.

Основу интенсификации процесса вычислений, как уже отмечалось выше, составляют: сокращение количества действий (операций, команд, обрабатываемых разрядов и т.д.) – фактор 1; сокращение времени на выполнение этих действий – фактор 2; совмещение во времени выполнения этих действий – фактор 3, а также ресурсосбережение за счет рационального использования оборудования – фактор 4.

В таблице 7.3 приведены методы, средства и факторы процесса интенсификации производства информационной продукции на основе совершенствования алгоритмического базиса.

Как видно из таблицы 7.3, процесс интенсификации при производстве информационной продукции во многом подобен материальному производству, хотя имеет некоторые особенности.

В конечном счете средства интенсификации процесса вычислений, приведенные в таблице 7.3, направлены на уменьшение временной скорости вычислений, а также алгоритмической сложности (сокращение объема программ и сокращение семантического разрыва между языками программирования высокого уровня и машинными инструкциями).

Интенсификация процесса вычислений

№ п/п	Методы и средства	Факторы процесса интенсификации				
		Фактор	Фактор	Фактор	Фактор	
		1	2	3	4	
1	2	3	4	5	6	
1	1. Оптимизация точности вычислений 1.1. Использование переменной точности (разрядности) счета 1.2. Минимизация константы асимптотики погрешности 1.3. Использование соответствующей нормы погрешности 1.4. Согласование норм погрешности начального приближения и итерационной формулы 1.5. Использование инкрементных вычислений 1.6. Обеспечение устойчивости счета 1.7. Уменьшение исходного интервала изменения аргумента					
2.	1. Увеличение скорости сходимости 1.1. Адаптивные по времени аппроксимации					
3	1. Использование прошлого труда (предыдущих либо заранее выполненных вычислений) 1.1. Табличные методы 1.2. Таблично-алгоритмические методы 1.3. Использование специальных вычислительных схем счета (Мотцкина, Белаги, Пана и других)					
4	Использование асинхронных вычислений (методы «цифра за цифрой» Чена, адаптивный асинхронный метод и т.д.)					

Продолжение таблицы 7.3

5	1.Использование непрерывных вычислений 1.1. Итерационные формулы 1.1.1. Методы «цифра за цифрой» 1.2. Рекуррентные соотношения 1.3. «Скользящие» вычисления (статистических характеристик, фильтрации сигналов и т.д.) 1.4. Динамические схемы счета 1.5.Адаптивные по данным аппроксимации	+		+*	+
			+		+
				+*	+
		+	+		+
		+	+		+
			+		
6	1.Использование рациональных схем счета 1.1. Типа БПФ 1.2. Винограда 1.3. Уолша, Горнера и др.	+		+*	+*
		+		+*	
		+		+*	+*
7	Использование рациональных видов аппроксимаций	+			
8	1. Экономические вычисления 1.1. Замена сложных операций простыми (экономичные аппроксимации с простыми коэффициентами) 1.2. Замена длинных данных короткими (п.1, 1.5)				
			+		
			+		
9	Использование эффективных схем счета на основе группировки и упорядочивания данных (методы вычисления статистических характеристик, обработки изображений, фильтрации и т.д., основанные на таблицах типа гистограмм)	+	+	+*	+
10	1. Использование вычислений с использованием обратных связей 1.1. Итерационные методы счета 1.2. Адаптивные по времени вычислений аппроксимации	+		+*	+
		+			
11	1. Эффективная организация вычислений 1.1. Использование RISC архитектуры 1.2. Использование CISC архитектуры 1.3. Использование интегрированной архитектуры			+*	
		+		+*	+*
			+	+*	+

Продолжение таблицы 7.3					
12	1. Эквивалентное преобразование алгоритмов и программ 1.1. Преобразования алгоритмов, направленные на соответствие используемому оборудованию 1.2. Преобразования, направленные на сокращение используемых ресурсов	+	+	+*	+
13	Уменьшение размерности исходной задачи, алгоритма (методы декомпозиции, методы понижения)				+*
14	1. Адаптация алгоритмов к условиям применения 1.1. Использование адаптивных аппроксимаций 1.2. Использование разложений по невязкам для параллельных вычислений 1.3. Распараллеливание алгоритма 1.4. Использование базисных последовательностей рекуррентных соотношений 1.5. Использование базы знаний, основанной на порождающих и конкурирующих алгоритмах	+			
		+		+	
				+	+
		+	+	+	+*
15	1. Устранение «узких мест» в процессе вычислений 1.1. Метод баланса (равнопрочности)				
			+		+

* – возможно использование при соответствующей структуре вычислительных средств.

В табл. 7.4 приведены данные об ожидаемом ускорении вычислений при использовании некоторых средств интенсификации процесса вычислений. В таблице 7.4 обозначены данные, взятые из работ [8, 12, 21 – 25].

Ожидаемое ускорение обработки информации

№ n/n	Направление оптимизации средств интенсификации	Ожидаемое ускорение обработки информации
1	Технологическая база аппаратных средств системы*	$10^3 - 10^5$
2	Общая организация системы в целом * RISC *	2 - 10
3	Рациональное построение программного обеспечения*	2 - 10^2
4	Анализ и оптимизация алгоритмов обработки изображений*	10 - 10^2
5	Применение эффективных эвристических методов решения задач обработки изображений*	10 - 10^3
6	Использование схемы БПФ для 1024 точек: с плавающей точкой*	10^2
	с фиксированной точкой*	10^3
7	Использование динамического режима счета функций для массива из 1024 точек	$5 \cdot 10^3 - 10^4$
8	Использование сегментной аппроксимации, состоящей из N сегментов	N
9	Использование табличных методов (по сравнению с операцией сложения)	10
10	Аппаратная поддержка, включая использование спецпроцессоров*	5 - 10^2
11	Конвейеризация вычислений (с n звеньями конвейера при полной загрузке)*	n
12	Ускорение сходимости за счет использования разложения по невязкам по сравнению с базовыми методами (порядка)	2^n
13	Вычисление медианы матрицы элементов $n \times n$ с использованием обобщенных таблиц (порядка)	n^2
14	Применение Чебышевского двухшагового циклического метода для решения разностных уравнений 500-го порядка по сравнению с методом Ричардсона*	4
15	Применение минимаксных методов приближения по сравнению с разложением в ряд Тейлора*	2^n

Средства, связанные с совершенствованием алгоритмического базиса, могут развиваться неограниченно и имеют направленность на интенсификацию процесса вычислений, что делает их действенным инструментом в решении проблемы производительности.

Среди этих средств важную роль играют адаптивные алгоритмы. Как было показано выше, роль этих алгоритмов важна в настоящее время и еще более – в обозримом будущем.

7.4. Адаптация и ее роль в процессе интенсификации вычислений

На современном этапе развития науки и техники необходим постоянный научно-технический прогресс, который во многом зависит от количества и качества информации в каждой конкретной прикладной области. При этом необходимо получать информацию более высокого качества, производить ее быстрее и дешевле, в больших объемах и эффективно ее использовать.

Современные ЭВМ и ВС являются сложными системами, которые образуются не простым механическим объединением составных частей, а в значительной степени их взаимодействием между собой, которое осуществляется на уровне технических средств, организационного, информационного и алгоритмического базисов.

Оптимизация проектируемых ЭВМ и ВС основывается, с одной стороны, на согласовании структуры системы и технических характеристик с режимами функционирования системы, а с другой стороны, с требованиями пользователя.

В последнее время наблюдается заметное взаимное влияние средств вычислительной техники на алгоритмическое обеспечение и наоборот.

С одной стороны, необходим определенный баланс между техническими средствами и алгоритмами; в идеальном случае каждой вычислительной структуре отвечает свой эффективный алгоритм. Верно и обратное. Такой баланс сокращает время счета и позволяет осуществить экономию используемых ресурсов.

С другой стороны, бурный рост числа специализированных и проблемно-ориентированных процессоров, а также возможность создания в начале следующего тысячелетия Гигачипов, содержащих миллиарды вентиляей, стимулируют разработку новых методов, алгоритмов и подходов их использования.

Помимо этого, в ближайшем будущем ожидается появление «эволюционирующих» программ и генетических алгоритмов, которые смогут сами себя совершенствовать в течение жизни [5].

Известно, что магистральным путем повышения производительности вычислительных средств в настоящее время является разработка многопроцессорных ВС. Для сбалансирования требований по производительности таких ВС и стоимости в их состав включают специализированные процессоры. Кроме это-

го, в персональные компьютеры также включают специализированные сопроцессоры для проведения вычислений с плавающей точкой и вычисления элементарных функций.

При этом как для спецвычислителей, так и универсальных ЭВМ и систем, требуется многообразие эффективных алгоритмов, обладающих определенными свойствами. Особо это важно для алгоритмов, имеющих массовое применение, т.е. алгоритмов нижнего уровня обработки. Проблема получения и использования эффективных алгоритмов усугубляется наличием большой номенклатуры процессоров, ЭВМ и ВС. Решение этой проблемы связано с адаптацией вычислительных и алгоритмических средств.

Как уже отмечалось выше, имеется сильная зависимость производительности вычислительных средств от соответствия алгоритмов решаемых задач их структуре. Поэтому решение проблемы адаптации непосредственно влияет на повышение производительности ЭВМ и ВС.

Проблема адаптации может быть решена на основе разных подходов. Один из подходов – разработка вычислительных средств с программируемой архитектурой и структурой. Но с точки зрения производительного использования машинного времени такие системы не лишены недостатков. Имеется большой класс машин, не обладающих свойствами реконфигурации, где предпочтительным является адаптация алгоритмов к структуре вычислительных средств. Однако гораздо больших успехов при решении этой проблемы можно достичь при взаимной адаптации алгоритмических и технических средств. В частности, на переходном этапе развития вычислительной техники можно использовать синтетические архитектуры, сочетающие мощные команды с RISC инструкциями с одновременным использованием адаптивных алгоритмов.

Основной эффект от использования адаптации в рассматриваемых выше случаях состоит в интенсификации (ускорении) процесса вычислений в сочетании с разумной достаточностью в использовании необходимых ресурсов.

В общем случае существуют два типа адаптации – статическая и динамическая.

Статическая адаптация соответствует специализированным вычислительным средствам и осуществляется заранее. Динамическая адаптация осуществляется в процессе решения задачи и поэтому больше соответствует универсальным и проблемно-ориентированным вычислительным устройствам.

Статическая адаптация, которая обеспечивает соответствие алгоритма вычислительной структуре, одновременно включает механизм выявления и устранения «узких мест», заранее планирует процесс вычислений, заранее вычисляет данные, коэффициенты аппроксимирующих выражений и т.д. Она не требует наличия адаптивного элемента внутри алгоритма.

В отличие от статической, динамическая адаптация требует наличия адаптивных элементов в самом алгоритме и/или в структуре вычислительных средств.

Но в том и другом случае адаптация выступает в роли средства интенсификации процесса вычислений. Эта интенсификация со стороны алгоритмического базиса достигается путем сокращения количества необходимых действий, минимизации допустимой погрешности, увеличения скорости сходимости, уменьшения константы асимптотики погрешности метода, обеспечения устойчивости счета, более полной загрузки устройств обработки, получения специализированных эффективных алгоритмов из обобщенных, организации динамических вычислений и т.д. Классификация алгоритмов по механизмам адаптации приведена в п. 7.2.

Необходимо отметить, что даже при разработке интенсивных средств ускорения вычислений также существует проблема интенсификации. Так, для простейшего метода статической адаптации – табличного, необходимо решить проблему – сокращение времени счета таблиц больших объемов с обеспечением необходимой точности.

Важность сокращения времени вычислений станет ясной из следующего примера [26].

При табличной реализации вычисления функции n – разрядного аргумента требуется предварительно вычислить q^n ее значений, где q – основание системы счисления. При $n=64$, $q=2$ и времени вычисления одного значения около 1 мкс – для вычисления $2^{64} = 1,85 \cdot 10^{19}$ значений необходимо $5,8 \cdot 10^5$ лет! Поэтому практические возможности прямых табличных методов ограничены небольшими разрядностями операндов и понятна важность использования адаптивных алгоритмов для обеспечения процесса интенсификации.

Так, использование адаптивных аппроксимаций [22, 27 – 32] для этих целей дает ускорение на 3 – 4 порядка на 1000 точек.

Аналогично обстоит дело и с таблично-алгоритмическими методами. Хотя объем таблицы в этом случае значительно сокращается, но «алгоритмическая часть» используется каждый раз, когда необходимо вычислить функцию. Поэтому и в данном случае использование адаптивных аппроксимаций для ускорения счета «алгоритмической части» также является крайне важным.

В вышеупомянутых случаях можно использовать динамические формулы счета, которые учитывают особенности вычисляемой функции и/или предшествующие им значения.

Таким образом, адаптивные методы позволяют сочетать универсальность и специализацию, а также адаптироваться к внешним и внутренним условиям применения. Эти особенности при правильном их использовании в конечном счете приводят к интенсификации процесса вычисления.

Имеется еще одно важное свойство адаптивных алгоритмов, связанных с режимами их использования (of-line и on-line). Особо это важно для использования алгоритмов в системах реального времени, когда вслед за поступлением текущих данных необходимо с минимальным запаздыванием выдать текущий результат.

Таким образом, адаптация вычислительных средств к внешним и внутренним условиям использования приводит к повышению эффективности функционирования этих средств.

7.5. Адаптивные аппроксимации и интенсификация процесса вычислений

Как было показано выше, одной из центральных проблем в ускорении процесса вычислений является обеспечение соответствия используемого алгоритма структуре вычислительных средств. Это может быть достигнуто без использования эффективных аппроксимаций вообще и адаптивных аппроксимаций, в частности.

Важность аппроксимаций при решении научных и прикладных задач связана с необходимостью замены сложных объектов исследования и применения более простых. Поэтому отнюдь не кажутся странными слова Нобелевского лауреата Б. Рассела: «Хотя это может показаться парадоксальным, вся наука подчинена идее аппроксимации» [33].

Статическая или динамическая адаптация алгоритма к внутренней структуре вычислительных средств, а также выбор наиболее оптимального алгоритма для данного применения позволяют ускорить процесс вычислений.

Хотя адаптивная аппроксимация (особо это относится к статической адаптации) имеет достаточно древнюю историю, например, использование таблиц, но основные результаты получены сравнительно недавно. Это относится прежде всего к таблично-алгоритмическим методам, методам сегментной и сплайн-аппроксимации. Большое влияние на развитие этих методов оказывал и оказывает в настоящее время уровень развития вычислительных средств. Не лишне упомянуть, что английский ученый Чарльз Беббидж в XVIII веке предложил проект «аналитической машины» для счета различного вида таблиц функций.

Механизмы динамической адаптации в основном отсутствуют. Поэтому понятна важность теории получения адаптивных по точности, сходимости, времени счета данных и т.д. аппроксимаций, которые излагаются в данной работе; предложены подходы для создания научных основ такой теории.

Благодаря этому в значительной степени снимаются противоречия между эффективными специализированными методами и универсальными, так как

эффективные специализированные методы могут быть получены из общих адаптивных методов, что делает их более обоснованными.

Получаемые адаптивные методы, рассматриваемые в работе, основываются на обобщенных базовых последовательностях ИФ, разложениях в ряды Тейлора-Маклорена, цепных дробях, аппроксимациях Паде, разложениях по ортогональным многочленам, минимаксных аппроксимациях и т.д. Адаптивные аппроксимации дают возможность выбрать в каждом конкретном случае наиболее подходящий для данного применения алгоритм вычисления. Основу этих методов составляют: функциональные преобразования и уравнения, переход от явного представления функций к неявному, операции над функциональными рядами и т.д., а также применение широкого спектра различных методов аппроксимаций.

Список литературы

1. Брюхович Е.И. Экономическая стратегия разработки вычислительных средств: место и роль счислений // УсиМ. – 1990. – № 2. – С. 3 – 18.
2. Брюхович Е.И. Стратегия разработки вычислительных средств и сетей с позиций экономических интересов их владельцев // УсиМ. – 1989. – № 3. – С. 3 – 14.
3. Крылов А.Н. Собрание трудов. Математика. – М.: Изд-во АН СССР, 1949. – 493 с.
4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Наука, 1987. – 600 с.
5. Райс Дж. Матричные вычисления и математическое обеспечение. – М.: Мир, 1984. – 264 с.
6. Дорфман В.Ф., Иванов Л.В. ЭВМ и ее элементы. Развитие и оптимизация. – М.: Радио и связь, 1988. – 240 с.
7. Смирнов А.Д. Архитектура вычислительных систем. – М.: Наука, 1990. – 320 с.
8. Мельников В.А., Дадаев Ю.Г. Супер –ЭВМ: Проблемы создания, использования и развития // Кибернетика. Становление информатики. – М.: Наука, 1986. – С. 70 – 90.
9. Бейт Р.Т. На пороге нового поколения транзисторов. В мире науки: Пер. с англ. – 1988. – № 5. – С. 66 – 71.
10. Квazar-микро inform. – 1994. – № 5 – 6. – С. 7.
11. Цифровая обработка информации на основе быстродействующих БИС / Гамкерлидзе С.А., Завялов А.В., Мальцев П.П., Соколов В.Г. – М.: Энергоиздат, 1988. – 136 с.
12. Энергомагнитные поля в триаде Человек – Земля – Вселенная / Бритиков А.А., Гвоздев В.И., Кузаев В.А., Спиридонов О.П. и др. // Зар. радиоэлектр. – 1995. – № 2 – 3.
13. Мисилович Н. Суперкомпьютеры в мире и нашей стране // Журнал д-ра Добба. – 1991. – № 3. – С. 12 – 15.
14. Матов В.Н., Белоусов Ю.А., Федосеев Е.П. Бортовые цифровые вычислительные машины и системы. – М.: Высшая школа, 1988. – 216 с.
15. Гимельфарб Г.Л. Аппаратные средства и особенности программного обеспечения диалоговой цифровой обработки изображений // Зарубежная радиоэлектроника. – 1985. – № 10. – С. 87 – 128.
16. Воеводин В.В. Математические модели и методы в параллельных процессорах. – М.: Наука, 1986. – 296 с.
17. Попов Б.А., Теслер Г.С. Вычисление функций на ЭВМ: Справочник. - Киев: Наукова думка, 1984. – 600с.
18. Иванов В.В. Методы вычислений на ЭВМ: Справочное пособие. - Киев: Наукова думка, 1986. – 584 с.

19. *Функционально ориентированные процессоры* / А.И. Водяхо, В.Б. Смолов, В.У. Плюснин, Д.В. Пузанков. – Л.: Машиностроение, 1988. – 224 с.
20. *Задирака В.К.* Теория вычисления преобразования: Фурье. – Киев: Наука думка, 1983. – 216 с.
21. *Попов Б.А.* Равномерное приближение сплайнами. – Киев: Наукова думка, 1989. – 272 с.
22. *Байков В.Д., Смолов В.Б.* Специализированные процессоры: Итерационные алгоритмы и структуры. – М.: Радио и связь, 1985. – 288 с.
23. *Благовещенский Ю.В., Теслер Г.С.* Вычисление элементарных функций на ЭВМ. – Киев: Техніка, 1977. – 208 с.
24. *Теслер Г.С.* Табулирование элементарных функций на ЦВМ // Алгоритмы и программы для вычисления функций на ЭВМ. – Киев: ИК АН УССР, 1972. – Вып. 1. – С. 127 – 137.
25. *Теслер Г.С.* Разложение функций в цепную дробь по невязкам // Алгоритмы и программы для вычисления функций на ЭВМ. – Киев: ИК АН УССР, 1976. – Вып. 3. – С. 117 – 125.
26. *Теслер Г.С.* Динамический режим вычисления функций на МВС с программируемой архитектурой на основе адаптивных алгоритмов // Многопроцессорные вычислительные структуры. – 1987. – Вып. 9. (XVIII). – С. 49 – 52.
27. *Теслер Г.С.* Адаптивные по данным алгоритмы вычисления некоторых элементарных функций для МКВС // Программное обеспечение ОВС / АН УССР ИПММС. Препринт № 5 – 90. – Львов. – 1990. – С. 38 – 43.
28. *Теслер Г.С.* Обобщенные адаптивные аппроксимации функций // Математичні машини і системи. – 1998. – № 2. – С. 3 – 8.
29. *Корнійчук Н.П., Лигун А.А., Доронин В.В.* Аппроксимация с ограничениями. – Киев: Наукова думка, 1982. – 252 с.