

НЕКОТОРЫЕ ПРОБЛЕМЫ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА НА СИСТЕМАХ ЭВМ

Ю. Г. Косарев (г. Новосибирск),
Е. Л. Ющенко (г. Киев)

Появление ЭВМ с большими объемами оперативной памяти (ОП) и вычислительных систем (ВС) заставляет по-новому взглянуть на разработку и состав математического обеспечения.

По мнению авторов настало время обсудить следующие вопросы.

1. Насколько и когда целесообразны мультипрограммные машины.

Мультипрограммный режим работы позволяет повысить коэффициент загрузки устройств ЭВМ. Выигрыш во времени при специально подобранных задачах обычно не превышает 1,5–2,0 раз, при валовой обработке задач значительно меньше. Этот выигрыш достигается за счет применения сложного диспетчера, занимающего заметную часть ОП и потребляющего известную долю времени.

Монопрограммный режим работы позволяет использовать для каждой задачи весь объем ОП. Как показал опыт математической эксплуатации ВС „Минск-222“, для многих задач при увеличении отводимого для них объема ОП можно значительно уменьшить время счета, применив более эффективные алгоритмы [1]. Для отдельных типовых задач увеличение ОП с 8.000 до 16.000 ячеек позволяет уменьшить время счета, например, при расчете подвески автомобилей в 1,9 раза; при некоторых расчетах атомных

реакторов - в 2,7 раза, в частности, при тепловом расчете реактора - в 30-35 раз. При расчете защиты реактора выигрыш еще больше. Заметим, что указанный выигрыш во времени определяется как отношение времени счета на одной машине ко времени счета на системе, умноженному на число машин в системе.

Значительный выигрыш во времени можно ожидать и для других классов задач. Для многих информационно-логических задач, например таких, как поиск информации, многопараметрическая обработка информации, упорядочение и т.п., увеличение ОП в ℓ раз сокращает время счета, как правило, в ℓ и более раз.

Обычные методы использования дополнительных объемов ОП сравнительно просты. Укажем основные из них:

применение табличных методов счета [2]. Кроме использования таблиц для многократно вычисляемых в данной задаче функций от одного, двух, а иногда и большего числа аргументов, зачастую оказывается выгодным отказаться от использования стандартных программ (СП), например $\cos x$, e^x , $\ln x$ и т.п., заменив их таблицами;

запоминание промежуточных данных, что позволяет избежать их повторного вычисления или записи и считывания из вспомогательных памятей;

для многих информационно-логических задач уменьшение числа обращений к внешним памятям и тем самым сокращение времени решения достигается благодаря увеличению размеров частей (и сокращения их числа) одного или нескольких массивов, вводимых в ОП в процессе решения задачи.

Нередко вся программа или ее основные блоки могут быть полностью размещены в ОП, что существенно упрощает программирование и делает ненужными различные усложнения, применяемые в ЭВМ с небольшими ОП.

Вывод: стремление выжать все возможное из мультипрограммности сейчас вряд ли можно считать

целесообразным. Видимо, достаточно ограничиться монопрограммным и двухпрограммным режимами работы.

2. О структуре вычислительной системы.

Имеются два основных пути:

1) построение ЭВМ с большой ОП;

2) построение нескольких однородных ЭВМ с тем же суммарным объемом ОП и объединение их в ВС.

На первом пути задачи программирования решаются проще, но зато хуже обстоит дело с надежностью, так как выход одного из блоков ведет к простоям всей ЭВМ.

Второй путь повышает надежность и увеличивает быстродействие (благодаря параллельной работе вычислителей), но несколько усложняет программирование. Однако, как показал опыт эксплуатации ВС „Минск-222“, эти усложнения невелики. Эффективная загрузка параллельно работающих вычислителей для многих задач может быть также обеспечена сравнительно просто [3].

Второй путь оказывается выгоднее согласно критерию стоимости операции В. М. Глушкова [4], так как стоимость вычислителя и схем управления составляет 10-20% общей стоимости современных ЭВМ.

3. Два основных подхода к программированию для ВС.

Первый подход - универсальный. За основу берется граф-схема связей между операторами, по которой выясняется, какие операторы могут выполняться независимо друг от друга. Затем эти операторы распределяются между машинами. Последнее осуществляется путем перебора (по крайней мере, авторам не известны работы, в которых бы это делалось иначе). Так как перебор приходится ограничивать сравнительно небольшим числом вариантов, то схемы получаются не наилучшими [5].

Второй—исходит из естественного членения задач на крупные блоки. За основу берутся те же структуры, которые используются для сокращения записи программы и упрощения процесса ее создания и использования, т. е. блочность и цикличность. Задача членится между машинами крупными блоками (чаще всего однородными), что осуществляется с помощью сравнительно простой методики распараллеливания по циклам [3].

Объем программы для системы при таком подходе оказывается не намного больше (10–20%), чем для одной машины. Простой машин, как правило, не превышают нескольких процентов от общего времени решения задачи [1]. Методика проверена на нескольких десятках задач различных типов.

4. Об автоматизации программирования для ВС.

Два основных пути:

1) разработка трансляторов для ВС, которые бы перерабатывали исходную программу для одной машины в программы для системы;

2) использование существующих языков и трансляторов и постепенное их совершенствование. Исходная программа при этом записывается с учетом схемы распараллеливания.

Второй путь на первых этапах может потребовать несколько больших усилий от составителя. Имеющиеся примеры программ для ВС „Минск-222” показывают, что объем записи алгоритмов больших задач увеличивается на 10–20%, а небольших — не более, чем вдвое [3].

На первом этапе особенно просто применение языков и трансляторов, которые допускают использование подпрограмм, записанных на машинном языке, либо непосредственно с указанием „код”, либо в виде библиотечных СП. Тогда функционирование системы может быть обеспечено несколькими стандартными процедурами.

На втором этапе в язык могут быть включены специальные средства, учитывающие особенности

функционирования системы. Это несколько упростит запись исходных программ и может улучшить результирующие программы.

На третьем этапе исходная программа записывается как и обычная программа для одной машины, но с указаниями о схеме распараллеливания.

Задача дальнейших этапов — упростить и свести к минимуму эти указания, т.е. второй путь не исключает первого.

Л и т е р а т у р а

1. Ю.Г.К о с а р е в, Опыт решения задач на системе „Минск-222“, Труды 1 Всесоюзной конференции по вычислительным системам, вып. 4, изд-во „Наука“, Сибирское отделение, г.Новосибирск, 1968.
2. Ю.Г.К о с а р е в, Примеры использования таблиц для сокращения времени счета, в сб. „Вычислительные системы“, вып. 30, изд-во „Наука“, Сибирское отделение, г.Новосибирск, 1968.
3. Ю.Г.К о с а р е в, Распараллеливание по циклам, в сб. „Вычислительные системы“, вып. 24, изд-во „Наука“, Сибирское отделение, г.Новосибирск, 1967.
4. В.М.Г л у ш к о в, Синтез цифровых автоматов, Физматгиз, М., 1962.
5. E.S.S c h w a r t z, An Automatik Sequencing Procedure with Application to Parallel Programming, J.Assoc. Compt. Machinery, 8, N 4, 1961, (Русский перевод — „Кибернетический сборник“, № 9, изд-во „Мир“, 1964).
6. Л.В.Г о л о в я ш к и н а, Ю.И.К о л о с о в а, Ю.Г.К о с а р е в, Н.Н.М я р е н к о в, Автоматизация программирования для систем на основе существующих трансляторов, в сб. „Вычислительные системы“, вып. 30, изд-во „Наука“, Сибирское отделение, г. Новосибирск, 1968.

Доложено на семинаре 22.У 1967 г.