

ПАРАЛЛЕЛЬНЫЕ СВОЙСТВА АЛГОРИТМА

Параллельные компьютеры (суперкомпьютеры) предназначены для быстрого решения больших задач. Чем мощнее компьютер, тем потенциально быстрее можно решить на нем задачу.

Помимо возможностей суперкомпьютера, скорость решения зависит также от алгоритма; например, если 50% операций алгоритма являются последовательными, то, согласно закону Амдала, сколько процессоров параллельного компьютера не используй, ускорение более чем в 2 раз получить не удастся. Поэтому надо изучать информационную структуру алгоритмов, уметь оценивать потенциальные возможности распараллеливания и создавать ресурс параллелизма. Кроме того, при использовании параллельного компьютера существенную роль играет степень согласованности структуры параллельного алгоритма с архитектурой компьютера. Поэтому получение параллельного алгоритма должно вестись с учетом типа целевого суперкомпьютера.

Таким образом, основными факторами, определяющими скорость выполнения алгоритма на параллельном компьютере, являются:

- возможности суперкомпьютера,
- ресурс параллелизма алгоритма,
- согласованность структуры параллельного алгоритма с архитектурой компьютера.

Использование многоядерной техники порождает проблемы вычислительной математики, которые не возникают при использовании последовательных компьютеров:

- получение информационной структуры алгоритмов,
- исследование параллельной структуры алгоритмов,
- построение параллельных алгоритмов, адекватных архитектуре целевого суперкомпьютера.

Способы представления зависимостей:

- аналитические: уровни зависимостей, векторы зависимостей, функции зависимостей (покрывающие функции), многогранники зависимостей;
- графовые: развернутый граф зависимостей (граф алгоритма), редуцированный граф зависимостей.

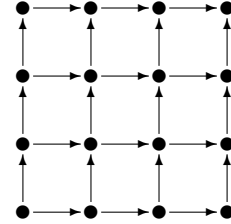
Доказано (теорема В.В. Воеводина об информационном покрытии), что все зависимости алгоритмов, представленных линейным классом программ, можно задать функциями, линейно зависящими от параметров циклов и внешних переменных. Число таких функций не зависит от внешних переменных и для реальных линейных программ пропорционально числу операторов

присваивания. Коэффициент пропорциональности не превосходит нескольких единиц. На меньшее число функций нельзя рассчитывать, если операторы связаны друг с другом.

Граф алгоритма (развернутый граф зависимостей) – это ориентированный граф, полученный следующим образом: каждая операция порождает вершину графа, каждая зависимость операции B от операции A порождает дугу, идущую из вершины, соответствующей операции A , в вершину, соответствующую операции B .

Приведем пример. Рассмотрим алгоритм, заданный следующим гнездом циклов:

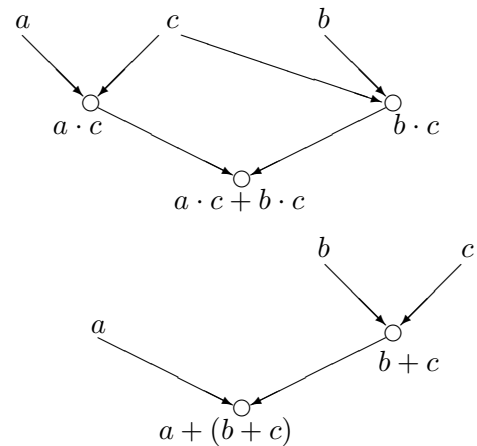
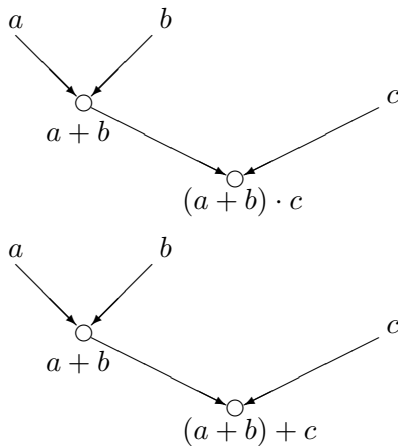
```
do i = 1, N
  do j = 1, N
    a(i, j) = a(i - 1, j) + a(i, j - 1)
  enddo
enddo
```



Поставим в соответствие каждой операции $a(i, j) = a(i - 1, j) + a(i, j - 1)$ алгоритма точку (i, j) . Зависимости между операциями изобразим стрелками. Получим граф алгоритма.

Граф алгоритма представляет информационное ядро алгоритма, дает полную количественную информацию о зависимостях алгоритма.

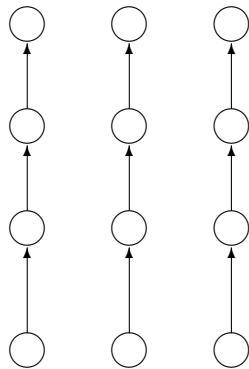
Приведем важное утверждение: в двух математически эквивалентных алгоритмах влияние ошибок округления одно и то же тогда и только тогда, когда графы алгоритмов изоморфны¹ (с учетом отличия входных данных и операций, приписанных вершинам). На рисунках изображены графы алгоритмов вычисления левых и правых частей равенств $(a + b) \cdot c = a \cdot c + b \cdot c$, $(a + b) + c = a + (b + c)$.



Параллельные множества. Пусть операции алгоритма разбиты на множества таким образом, что никакая операция из одного множества непосредственно или опосредованно не зависит от операций из других множеств.

¹Графы $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ называются изоморфными, если существует такая биекция $f : V_1 \rightarrow V_2$, что в G_1 есть ребро, соединяющее вершины v_i и v_j , тогда и только тогда, когда в G_2 есть ребро между $f(v_i)$ и $f(v_j)$.

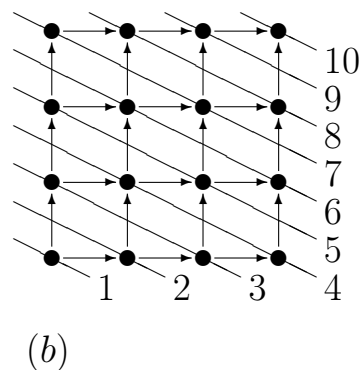
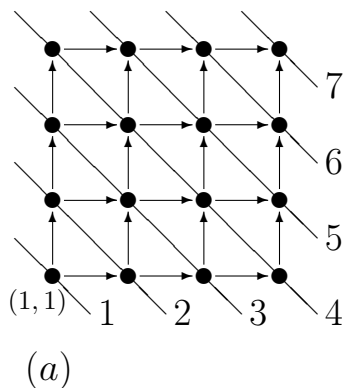
Такие множества операций называются параллельными. Параллельные множества операций алгоритма можно выполнять независимо друг от друга. Ниже приведен граф алгоритма (умножение матрицы порядка 3 на вектор), в котором три параллельных множества операций.



Существование параллельных множеств операций алгоритма означает наличие естественного параллелизма.

Параллельная форма алгоритма. Пусть операции алгоритма разбиты на группы, упорядоченные так, что каждая операция любой группы зависит либо от начальных данных алгоритма, либо от результатов выполнения операций, находящихся в предыдущих группах. Представление алгоритма в подобном виде называется параллельной формой алгоритма. Каждая группа операций называется ярусом параллельной формы, число групп – высотой параллельной формы, максимальное число операций в ярусе – шириной параллельной формы. Минимальная высота параллельных форм алгоритма называется высотой алгоритма. При выполнении алгоритма на многопроцессорной вычислительной системе всегда реализуется некоторая параллельная форма. Операции, принадлежащие одному ярусу, можно выполнять независимо друг от друга.

Параллельную форму алгоритма для рассмотренного выше примера можно построить, отнеся к ярусу с номером t все операции (i, j) , для которых $i + j - 1 = t$ (рис. (a), $N = 4$).



Высота параллельной формы равна $2N - 1$, ширина — N . Соответствующий параллельный алгоритм можно записать в следующем виде:

```

do  $t = 1, 2N - 1$ 
  dopar  $j = \max(1, t - N + 1), \min(t, N)$ 
     $i = t - j + 1$ 
     $a(i, j) = a(i - 1, j) + a(i, j - 1)$ 
  enddopar
enddo

```

Если к одному ярусу с номером t отнести все операции (i, j) , для которых $i + 2j - 2 = t$ (рис. (b)), то высота параллельной формы равна $3N - 2$, ширина (для четных N) равна $\frac{N}{2}$.

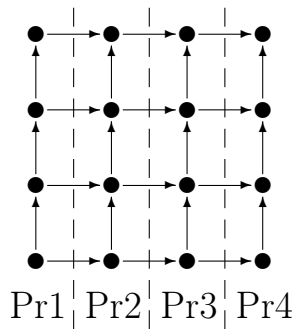
Если известна параллельная форма алгоритма, то сам алгоритм можно реализовать на параллельной вычислительной системе по шагам последовательно ярус за ярусом. Если число процессоров и соединений между ними достаточно, то алгоритм можно реализовать на параллельной системе за число шагов, равное высоте алгоритма.

Отметим, что если с помощью бинарных или унарных операций вычисляется значение некоторого выражения, существенным образом зависящего от N переменных, то высота алгоритма, позволяющего вычислить это выражение, не меньше $\log_2 N$.

Параллельные последовательности вычислений. Параллельные вычислительные процессы. Параллельные последовательности вычислений — это упорядоченные множества операций алгоритма, которые могут выполняться одновременно. Зависимыми могут быть как операции одной, так и операции разных последовательностей. Явного указания операций, выполняемых одновременно, не требуется. Параллельные вычислительные процессы — это процессы одновременного выполнения операций алгоритма. Зависимыми могут быть как операции, выполняемые одним процессом, так и операции, выполняемые разными процессами. Явного указания операций, выполняемых одновременно, не требуется.

Существование параллельных вычислительных процессов означает наличие внутреннего параллелизма.

Проиллюстрируем задание параллельных последовательностей вычислений с помощью рисунка.



Операции, приписанные точкам на рисунке, можно выполнять параллельно, если одна координата точек задает номер процесса, другая — порядок выполнения операций. После выполнения операции данные передаются процессу с

большим номером. Этот пример иллюстрирует задание конвейерного параллелизма на уровне операций алгоритма (или на уровне макроопераций, если вершинам графа приписаны множества операций).

Параллельные свойства алгоритма. В параллельных вычислениях наряду с традиционными вычислительными свойствами алгоритма — общее число выполняемых операций, влияние ошибок округления, размер необходимой памяти, локальность, сходимость — требуется исследовать новые, параллельные свойства: число параллельных множеств, высота и ширина параллельной формы, число параллельных вычислительных процессов, число и объем коммуникационных операций, сложность коммуникационных связей. Наличие одних хороших свойств у алгоритма или программы не гарантирует, что хорошими будут и какие-то другие свойства.

Параллельная структура алгоритма — совокупность сведений о параллельных множествах, параллельных формах алгоритма, параллельных процессах вычислений.