Шигаров Н.А, 242

**Лабораторная работа №2**

*Задача: «Построить модель вычислений в виде графа «операции-операнды» для нахождения евклидовой меры»*

Представим вычислительную модель алгоритма в виде графа "операции-операнды" в виде ациклического ориентированного графа 𝐺=(𝑉,E). 𝑉={1,…,|𝑉|} – множество вершин графа, представляющее выполняемые операции алгоритма, а *E* – множество ребер графа (при этом ребро *e = (i, j)* принадлежит графу только, если операция *j* использует результат выполнения операции *i*).

Операции алгоритма, между которыми отсутствует путь, могут выполняться параллельно. Пусть p – это число процессоров, используемых для выполнения алгоритма. Для параллельного выполнения вычислений необходимо задать множество 𝐻𝑝={(𝑖,𝑃𝑖,𝑡𝑖):𝑖𝜖𝑉}, где для каждой операции 𝑖𝜖𝑉 указывается номер используемого процессора 𝑃𝑖 и время начала выполнения операции 𝑡𝑖. Также потребуем выполнения следующих условий:

1) Один и тот же процессор не должен выполнять разные операции в одинаковый момент времени,

2) К назначаемому моменту выполнения операции все необходимые данные уже должны быть получены.

Модель вычислительной схема алгоритма 𝐺 вместе с расписанием 𝐻𝑝 можно рассматривать как модель параллельного алгоритма 𝐴𝑝(𝐺,𝐻𝑝), исполняемого с использованием p процессоров.

Время выполнения параллельного алгоритма определяется максимальным значением времени, используемым в расписании: 𝑇𝑝(𝐺,𝐻𝑝)=max{𝑖𝜖𝑉(𝑡𝑖+1) }

𝑇∞=min{𝑝≥1𝑇𝑝}

𝑇∞ - минимально возможное время выполнения параллельного алгоритма при использовании неограниченного количества процессоров.

𝑇1 - время выполнения алгоритма при использовании одного процессора

Для оценки эффективности параллельного решения величину 𝑇1 следует определять с учетом всех возможных последовательных алгоритмов 𝑇1=min𝐺{𝑇1(𝐺)}

**Теорема 1**. Пусть для некоторой вершины вывода в вычислительной схеме алгоритма существует путь из каждой вершины ввода. Кроме того, пусть входная степень вершин схемы не превышает 2. Тогда минимально возможное время выполнения параллельного алгоритма ограничено снизу значением

𝑇∞(𝐺)=log𝑛,

где *n* есть количество вершин ввода в схеме алгоритма.

Для определения времени выполнения заданного алгоритма при использовании одного процессора рассмотрим следующую последовательную схему нахождения евклидовой меры (синие кружки – iй элемент вектора x, красные – y):

Очевидно, сумм будет: 𝑛−1, и одна операция извлечения корня в конце. Таким образом, 𝑇1(𝐺)=n.

Для определения вычисления времени выполнения параллельного алгоритма рассмотрим каскадную схему нахождения евклидовой меры:

Cначала все элементы первого и второго массивов разбиваются на пары (получается 𝑛/2 пар, в случае если 𝑛 – нечетное, то добавляется ещё одна неполная пара). Далее в каждой из пар производится суммирование. На следующей ступени вычисляется сумма опираясь на посчитанные значения. В итоге мы получим 𝑛-1 сумму и 1 операцию извлечения корня в конце, т.е. столько же, сколько и при последовательном алгоритме, но поскольку для данной схемы можно распараллелить определенные операции сравнения, то время выполнения для этого алгоритма будет меньше (при этом 𝑝≥⌈𝑛⁄2⌉), а именно 𝑇𝑝(𝐺,𝐻𝑝)= log𝑛.

Время будет именно таким, поскольку данная вычислительная схема удовлетворяет теореме 1

Вычислим ускорение 𝑆𝑝(𝑛)= 𝑇1(𝑛)/𝑇𝑝(𝑛): Sp(n) = n / log n

Максимальное ускорение по Амдалу: 𝑆𝑝=n /[log n / n + (1 – log n) / p