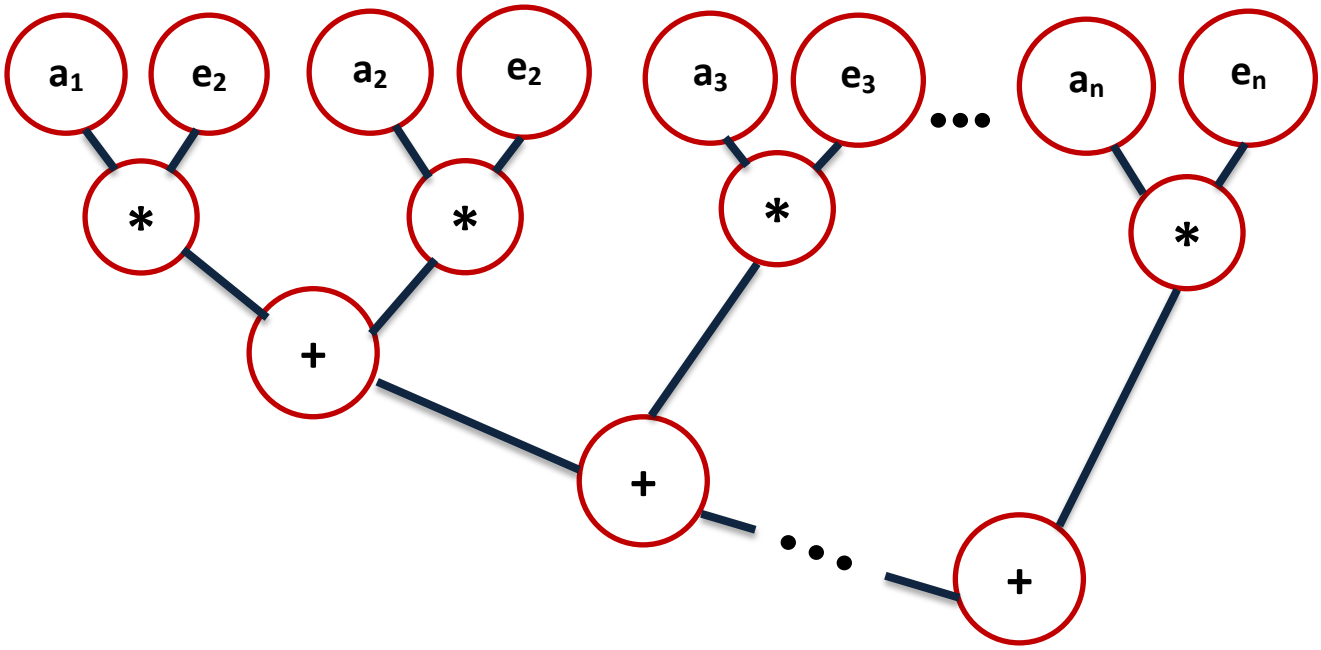
Лабораторная работа №4

СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ ВЕКТОРОВ

1. Задача: перемножить скалярно 2 n-мерных вектора.
Исходные данные: два одномерных массива n чисел.
Пусть даны 2 n-мерных вектора $\bar{a}(a_1, a_2,\ldots, a_n)$ и $\bar{e}(e_1, e_2,\ldots, e_n)$. Тогда их скалярное произведение вычисляется по формуле: $(\bar{a}, \bar{e}) = a_1e_{1+} a_2e_2+ \ldots +a_ne_n$.
Вычисляемые данные: сумма попарных произведений элементов массива.

Схема последовательного варианта решения*:



*Последовательность исполнения суммирования может быть разная — как по возрастанию, так и по убыванию индексов**.**

Для вычисления скалярного произведения массивов, состоящих из n элементов, при любых разложениях количество операций умножения неизменно и равно n, а количество операций сложения равно n - 1. Поэтому алгоритм должен быть отнесён к алгоритмам линейной сложности по количеству последовательных операций.

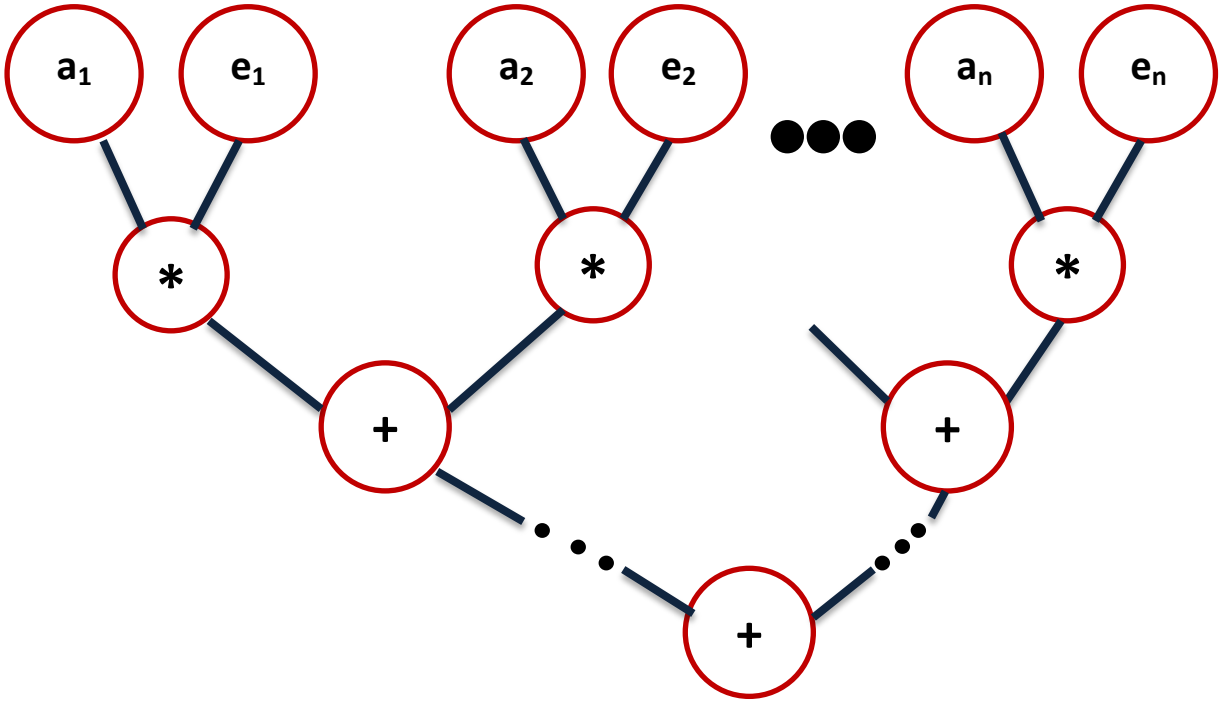
Сложность данного алгоритма: O(n).

2. Распараллелим. Для решения одной и той же задачи можно использовать разные схемы вычислений, одни из которых будут распараллеливаться лучше, чем другие. Задача построения оптимального параллельного алгоритма состоит в том, чтобы при построении модели вычислений выбрать наиболее подходящую для параллельного выполнения схему.
В случае если n не является степенью двойки, дополним n до ближайшей степени двойки нулевыми элементами.

Алгоритм распараллеливания: 1) перемножаем i-координаты векторов;
2)распределяем получившиеся произведения на двойки ($i^{ая}$ с $(i+1)^{ой}$ координатой;
3)складываем значения, находящиеся в одной двойке;
4)переименовываем координаты (от 1 до [n/2]);
5)если n>1, к пункту 3), иначе умножение выполнено.

3. Граф «операции – операнды» - это ациклический ориентированный граф, в котором вершины представляют выполняемые операции алгоритма, где V = {1, … ,N} – множество вершин графа, а R – множество дуг графа вида r_{ij}, где i,j ∈ V. Дуга графа r_{ij} направлена от вершины i к вершине j только, если операция j использует результат выполнения операции i.

Рассмотрим граф «операции – операнды» для алгоритма, вычисляющего скалярное произведение векторов $\bar{a}(a_1, a_2,\ldots, a_n)$ и $\bar{e}(e_1, e_2,\ldots, e_n)$, вычисляемого по формуле:
$(\bar{a}, \bar{e}) = a_1e_{1+}\, a_2e_2+ \ldots +a_ne_n.$



Операции алгоритма, между которыми нет пути в графе, могут быть выполнены параллельно.

4. Оценим ускорение и эффективность:

Ускорение параллельной программы (алгоритма), получаемое при запуске программы на системе с p процессорами, – это отношение $\mathbf{S_p = T_1 / T_p}$, где T_1– время выполнения программы на одном процессоре; T_p–время выполнения программы на системе из p процессоров.

Эффективность использования параллельной программой ресурсов многопроцессорной вычислительной системы при решении задачи определяется соотношением $\mathbf{E_p = S_p / p}$.

При создании высокопроизводительных программ необходимо стремиться к тому, чтобы S_p -> p, E_p -> 1.

Примем за единицу время выполнения операций.
Количество итераций каскадной схемы: $K = \log_2 n$
Количество операций суммирования: $K_{add} = n-1$
Количество операций умножения: $K_{mult} = n$

$T_1(n) \approx (t_{add} + t_{mult})*n = 2n$

Для каскадной схемы при $p \geq [n/2]$

$T_\infty(n) \approx (n/p)*(t_{add} + t_{mult}) + \log_2 p(t_{add} + t_{comm}) \leq 4 + 2\log_2 n$

$\mathbf{S_p} = T_1(n)/ T_\infty(n) \approx n/(2 + \log_2 n)$

$\mathbf{E_p} = T_1(n)/(p*T_\infty(n)) \approx 2n/((4 +2 \log_2 n)*p) = 2/(2+\log_2 n)$ -> 0 (при n -> infinity)

Определен минимальный диаметр среди всех возможных графов. $T_\infty = \log_2 n$.
Время выполнения алгоритма сопоставляется с минимально возможным.
p -> $2n/ \log_2 n$
$p \geq T_1 / T_\infty$ => $T_p \leq 2 T_\infty$

Значит, $T_p \leq 2 \log_2 n$

$S_p = (T_1(n) / T_p(n)) \geq 2n/(2 \log_2 n) = n/\log_2 n$

$E_p = T_1(n)/(p*T_p(n)) \geq n/(\log_2 n*(2n/ \log_2 n)) = ½$

Согласно закону Амдала, ускорение выполнения программы за счёт распараллеливания её инструкций на множестве вычислителей ограничено временем, необходимым для выполнения её последовательных инструкций.
Предположим, что алгоритм задачи таков, что доля α от общего объёма вычислений может быть получена только последовательными расчётами, а, соответственно, доля (1 – α) может быть распараллелена идеально (то есть время вычисления будет обратно пропорционально числу задействованных узлов p). Тогда ускорение, которое может быть получено на вычислительной системе из p процессоров, по сравнению с однопроцессорным решением не будет превышать величины

$S_p = 1/(\alpha+((1-\alpha)/p))$

Максимальное ускорение достигается в алгоритме, не содержащем последовательных операций (α=0).

$\mathbf{S_p} = 1/(0+((1-0)/(2n/ \log_2 n))) = 2n/\log_2 n$
Следовательно, $\mathbf{p_{max}} = 2n/ \log_2 n$ и $\mathbf{E_p}$ -> 1