

Оглавление

Определение 1. Ускорение, получаемое при использовании параллельного алгоритма для p вычислителей, по сравнению с последовательным вариантом выполнения вычислений, определяется величиной

$$S_p(n) = \frac{T_1(n)}{T_p(n)}$$

Определение 2. Эффективность использования параллельным алгоритмом вычислителей при решении задачи определяется соотношением

$$E_p(n) = \frac{T_1(n)}{pT_p(n)} = \frac{S_p(n)}{p}$$

Замечание. Сверхлинейное ускорение $S_p > p$ может иметь место в силу неравноправности выполнения последовательной и параллельной программ, нелинейного характера зависимости сложности от объёма обрабатываемых данных, и т. д.

Определение 3. Стоимость вычислений определяется как произведение времени параллельного решения задачи и числа используемых вычислителей

$$C_p = pT_p$$

Закон Амдаля (Amdahl)

Достижению максимального ускорения может препятствовать существование в выполняемых вычислениях последовательных расчётов, которые не могут быть распараллелены.

Пусть f — доля последовательных вычислений в применяемом алгоритме обработки данных.

Ускорение процесса вычислений при использовании p вычислителей оценивается величиной:

$$S_p = \frac{T_1}{T_p} = \frac{f + (1-f)}{f + \frac{1-f}{p}} \leq S^* = \frac{1}{f}$$

Закон Густавсона–Барсиса

Оценим максимально достижимое ускорение исходя из имеющейся доли последовательных расчётов в выполняемых параллельных вычислениях:

$$g = \frac{\tau(n)}{\tau(n) + \frac{\pi(n)}{p}},$$

где $\tau(n)$ и $\pi(n)$ — времена последовательной и параллельной частей, т. е.

$$T_1 = \tau(n) + \pi(n), \quad T_p = \tau(n) + \frac{\pi(n)}{p}$$

С учётом величины g можно получить

$$\tau(n) = g \left(\tau(n) + \frac{\pi(n)}{p} \right), \quad \pi(n) = (1-g)p \left(\tau(n) + \frac{\pi(n)}{p} \right),$$

что позволяет получить оценку для ускорения

$$S_p = \frac{T_1}{T_p} = \frac{\tau(n) + \pi(n)}{\tau(n) + \frac{\pi(n)}{p}} = \frac{\left(\tau(n) + \frac{\pi(n)}{p}\right) \left(g + (1-g)p\right)}{\tau(n) + \frac{\pi(n)}{p}}$$

Определение 4. Параллельный алгоритм называют *масштабируемым*, если при росте числа вычислителей он обеспечивает увеличение ускорения при сохранении постоянного уровня эффективности использования процессоров.

Определение 5. *Накладные расходы* при выполнении параллельного алгоритма появляются за счёт необходимости организации взаимодействия вычислителей, выполнения некоторых дополнительных действий, синхронизации параллельных вычислений, и т. д.

Пусть $E = \text{const}$ — желаемый уровень эффективности выполняемых вычислений. Из выражения для эффективности можно получить

$$\frac{T_0}{T_1} = \frac{1-E}{E}, \quad T_1 = \frac{E}{1-E} \cdot T_0$$

Порождаемую этим соотношению зависимость $n = F(p)$ обычно называют *функцией изоэффективности*.