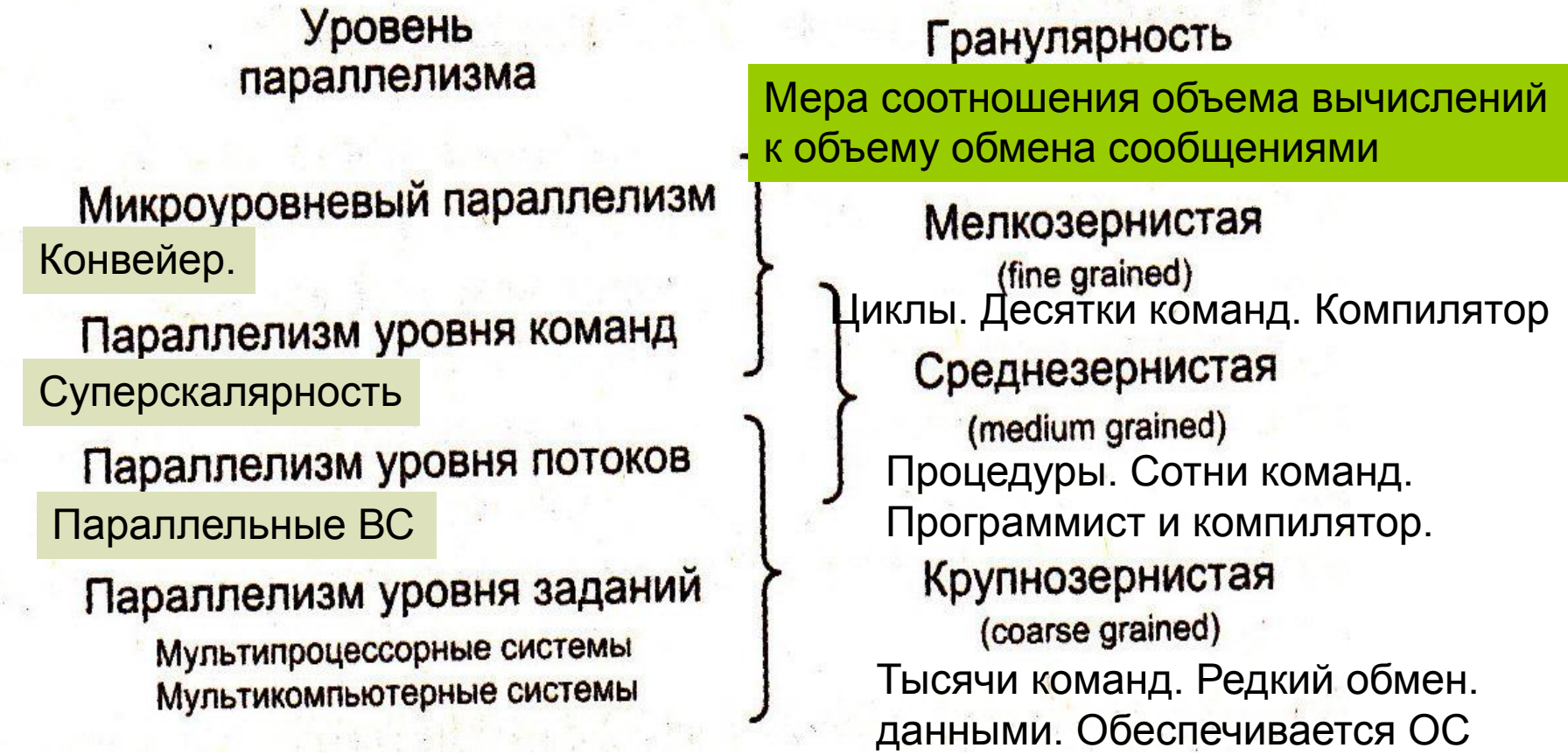
Метрики параллельных вычислений

Цель лекции: рассмотреть теоретические основы построения МП, реализующих различные уровни параллелизма вычислений

Три вопроса разработки компьютеров параллельного действия

- 1. Каков тип, размер и количество процессорных элементов?
- 2. Каков тип, размер и количество элементов памяти?
- 3. Как взаимодействуют элементы памяти и процессорные элементы?

Уровни параллелизма



Гранулярность и коммуникационная задержка

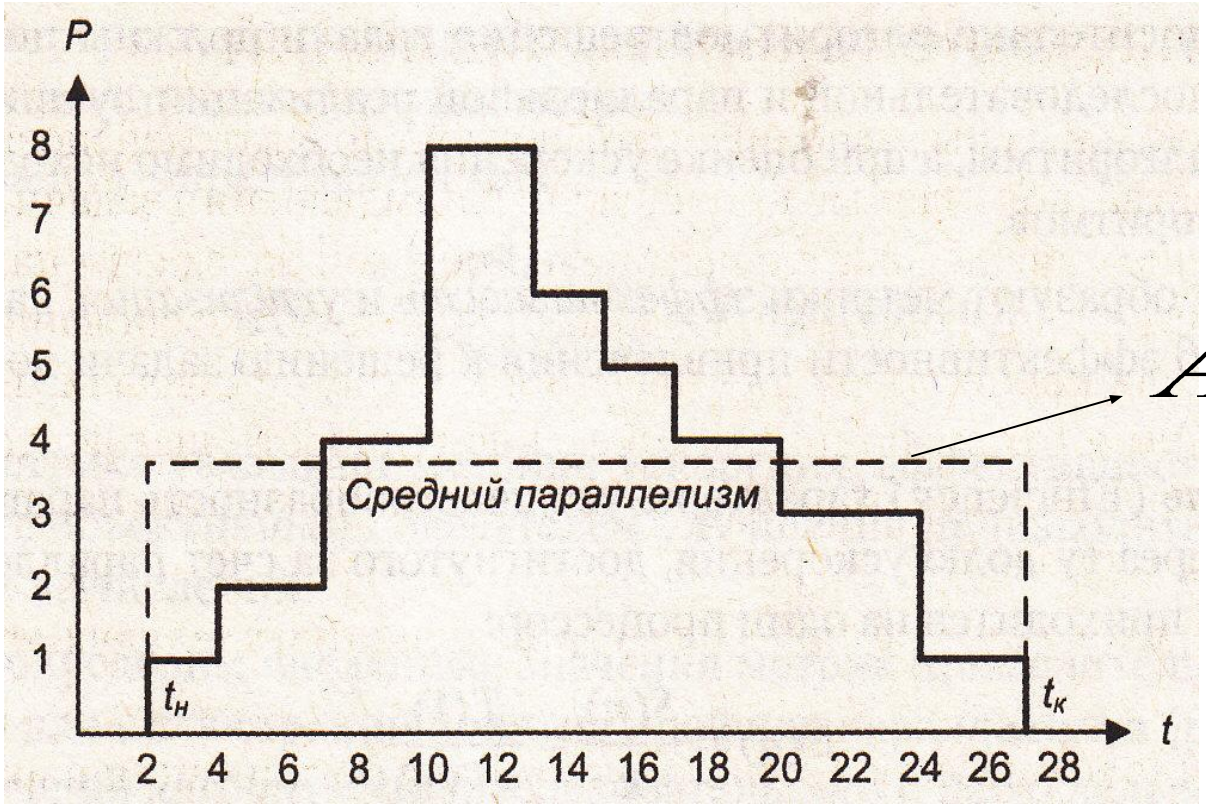
- Если коммуникационная задержка минимальна, то наилучшую производительность обещает мелкоструктурное разбиение программ.

- Если коммуникационная задержка велика, то предпочтительней крупнозернистое разбиение программ.

Профиль параллелизма программы

Графическое представление P(t) называют профилем параллелизма.

Средний параллелизм программы



$$A = \frac{\sum_{i=1}^{n} i t_i}{\sum_{i=t}^{n} t_i}$$

A=(1*5+2*3+3*4+4*6+5*2+6*2+8*3)/(5+3+4+6+2+2+0+3) = 93/25 = 3.72

Метрики параллельных вычислений

Это система показателей, позволяющих оценить преимущества, получаемые при параллельном решении задачи на N- процессорах.

n – количество процессоров.

O(n) – объем вычислений, через количество операций выполняемых n.

T(n) – общее время вычислений с использованием n процессоров.

Индекс параллелизма PI(n)

- Характеризует среднюю скорость параллельных вычислений через количество выполненных операций.

$$PI(\mathrm{n}) = O(\mathrm{n})/\mathrm{T}(\mathrm{n})$$

Объем вычислений

Время вычислений n - процессорами

Ускорение

- Ускорение за счет параллельного выполнения программы – показатель эффективности скорости вычислений. Вычисляется как отношение времени на проведение вычислений однопроцессорной системой, ко времени решения той же задачи на параллельной n-процессорной системе.

$$S(n) = T(1)/T(n)$$

Эффективность

- Характеризует целесообразность наращивания числа процессоров через ту долю ускорения, достигнутого за счет параллельных вычислений, которая приходится на один процессор

Ускорение Время вычисления однопроцессорной ВС

$$E(\mathrm{n}) = \frac{S(\mathrm{n})}{\mathrm{n}} = \frac{T(1)}{\mathrm{n}T(\mathrm{n})}$$

Утилизация

- Учитывает вклад каждого процессора при параллельном вычислении в виде количества операций, выполненных процессором в единицу времени.

$$U(\mathrm{n}) = \frac{\mathrm{O(n)}}{\mathrm{nT(n)}}$$

Избыточность

- Отношение объема параллельных вычислений к объему последовательных вычислений.

$$R(n) = \frac{O(n)}{O(1)}$$

Сжатие

- Величина обратная избыточности.

$$C(n) = \frac{O(1)}{O(n)}$$

Качество – обобщающий показатель

- Данная метрика связывает метрики ускорения, эффективности и сжатия и является обобщающим показателем улучшения производительности за счет параллельных вычислений.

$$Q(n) = S(n)E(n)C(n)$$

Пример применения метрик

Пусть наилучший алгоритм для последовательного и параллельного вычисления совпадают и дано n=8; T(1)=O(1)=O(8)=93; T(8)=25

Индекс параллелизма

$$Pl(8) = \frac{O(8)}{T(8)} = \frac{93}{25} = 3,72;$$

ускорение

$$S(8) = \frac{T(1)}{T(8)} = \frac{93}{25} = 3,72;$$

эффективность

$$E(8) = \frac{T(1)}{8T(8)} = \frac{93}{8 \times 25} = 0,465;$$

утилизация

$$U(8) = \frac{O(8)}{8T(8)} = \frac{93}{8 \times 25} = 0,465;$$

избыточность

$$R(8) = \frac{O(8)}{O(1)} = \frac{93}{93} = 1;$$

сжатие

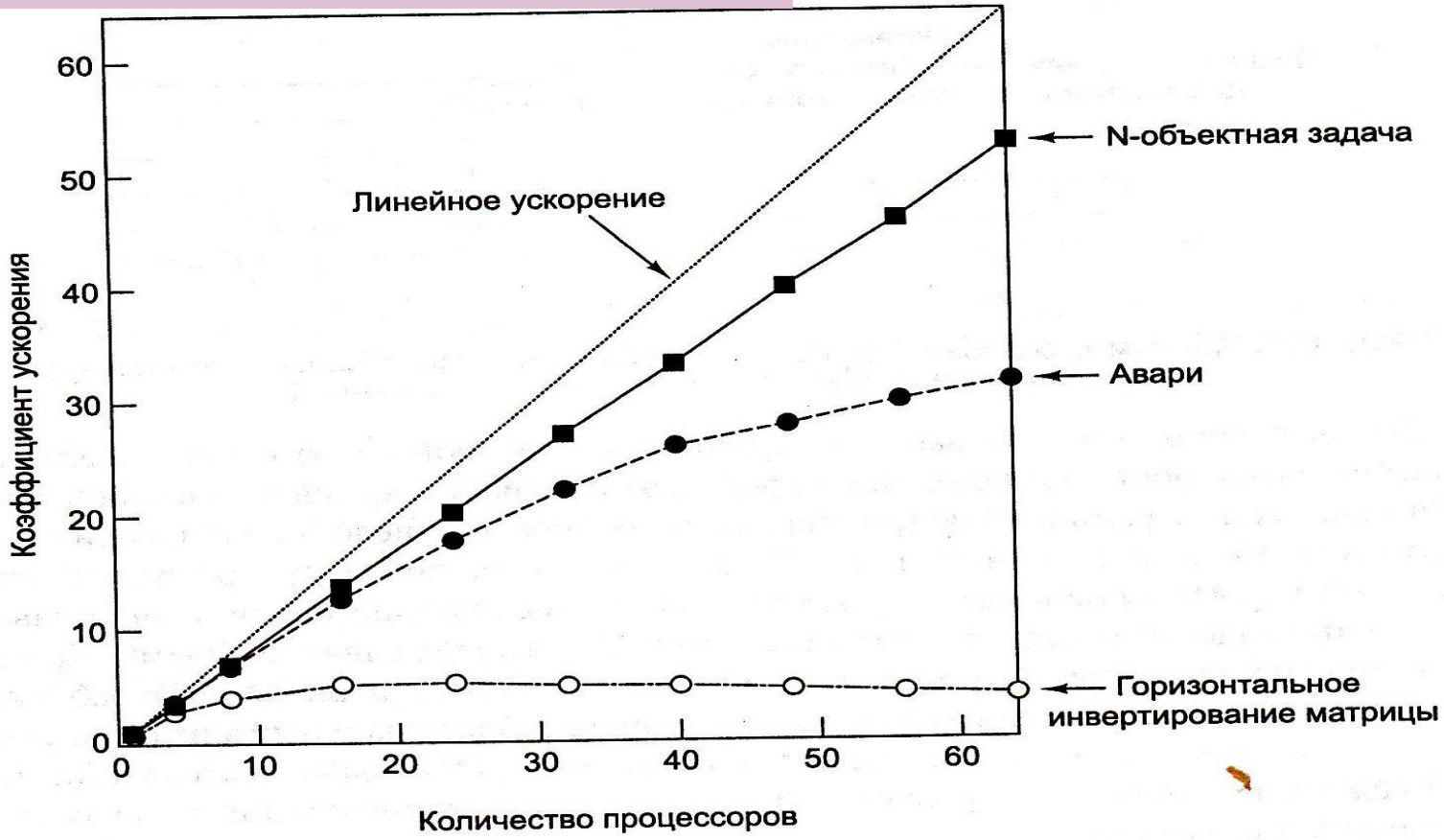
$$C(8) = \frac{O(1)}{O(8)} = \frac{93}{93} = 1;$$

качество

$$Q(8) = S(8)E(8)C(8) = 3,72 \times 0,465 \times 1 = 1,73.$$

Идеальное и реальное ускорение

Зависимость от алгоритма программы



Самый главный вопрос – Какое можно получить ускорение при увеличении количества процессоров?

Причины недостижимости идеального ускорения

- Все программы имеют последовательную часть, которая не может быть распараллелена:
- фаза инициализации;
- фаза считывания данных;
- фаза сбора результатов.
- Фаза коммуникации

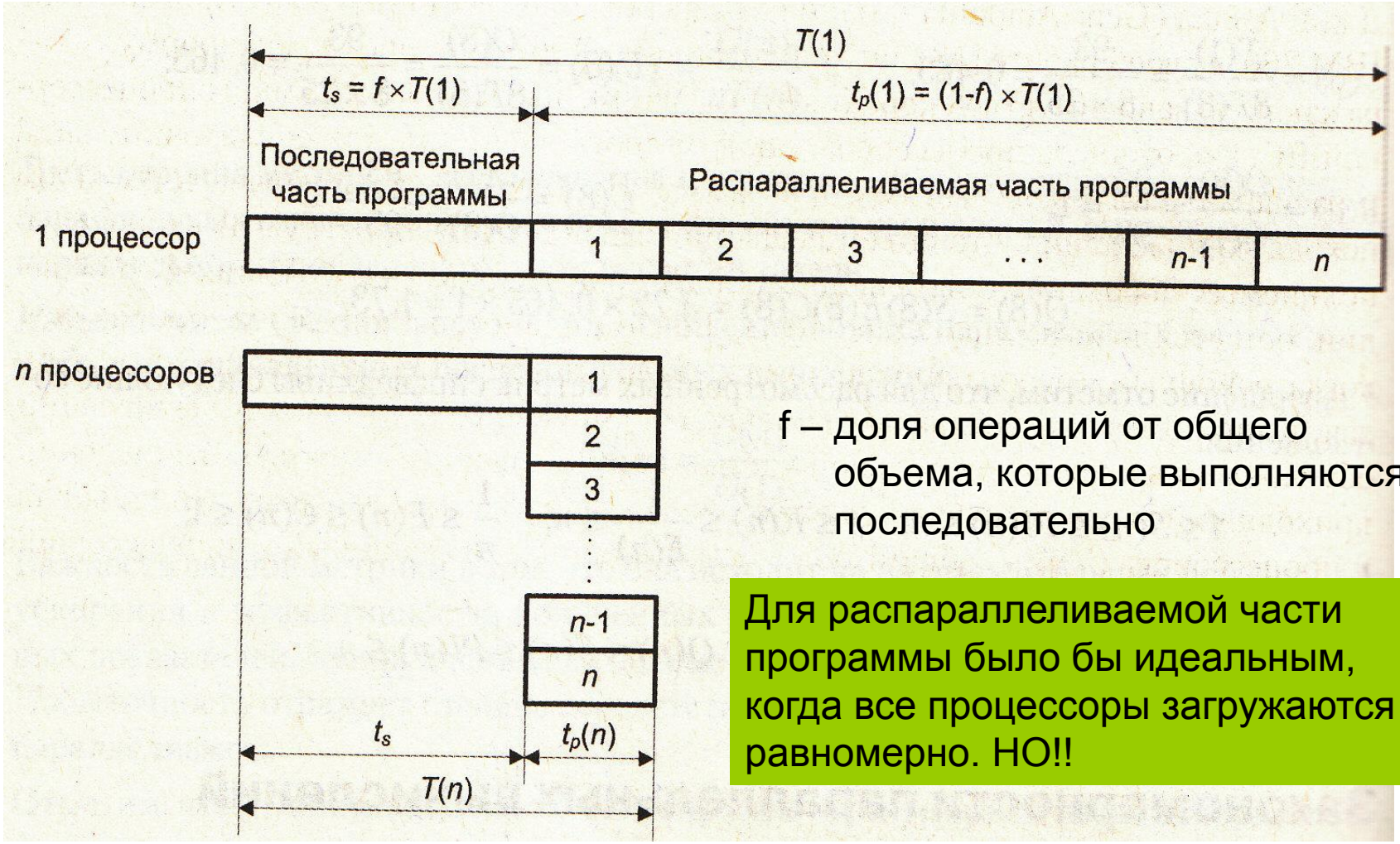
Причины недостижимости идеального ускорения

- **Издержки из-за дисбаланса загрузки процессоров.** Между точками синхронизации каждый из процессоров должен быть загружен одинаковым объемом работы, иначе часть процессоров будет ожидать, пока остальные завершат свои операции. Эта ситуация известна как *дисбаланс загрузки.*
- Таким образом, ускорение ограничивается наиболее медленным из процессоров.

Причины недостижимости идеального ускорения

- **Коммуникационные издержки.** Если принять, что обмен информацией и вычисления могут перекрываться, то любые коммуникации между процессорами снижают ускорение. В плане коммуникационных затрат важен уровень гранулярности, определяющий объем вычислительной работы, выполняемой между коммуникационными фазами алгоритма. Для уменьшения коммуникационных издержек выгоднее, чтобы вычислительные гранулы были достаточно крупными и доля коммуникаций была меньше.

Закономерности параллельных вычислений



f – доля операций от общего объема, которые выполняются последовательно

Для распараллеливаемой части программы было бы идеальным, когда все процессоры загружаются равномерно. НО!!

Дополнительные факторы влияющие на ускорение

- Время ожидания в коммуникациях.
- Ограниченная пропускная способность каналов.
- Недостатки алгоритмов программной реализации.

Три закона параллельных вычислений

- Главный Вопрос – на какое реально ускорение можно рассчитывать при увеличении количества процессоров?
- Ответ – все зависит от того как пользователь распорядится увеличенной мощностью. Наиболее характерными являются три ситуации.

1 ситуация	2	3 ситуация
Объем вычислений не меняется, цель – сократить время вычислений	Время вычислений не меняется, но увеличивается объем. Цель – в заданное время выполнить максимальный объем вычислений.	Как вторая, но емкость доступной памяти ограничена
Закон Амдала	Закон Густафсона	Закон Сана-Ная

Закон Амдала

IBM

Джин Амдал предложил формулу, отражающую зависимость ускорения вычислений, от числа процессоров и от соотношения последовательной и распараллеливаемой частей программы. Объем вычислений не меняется.

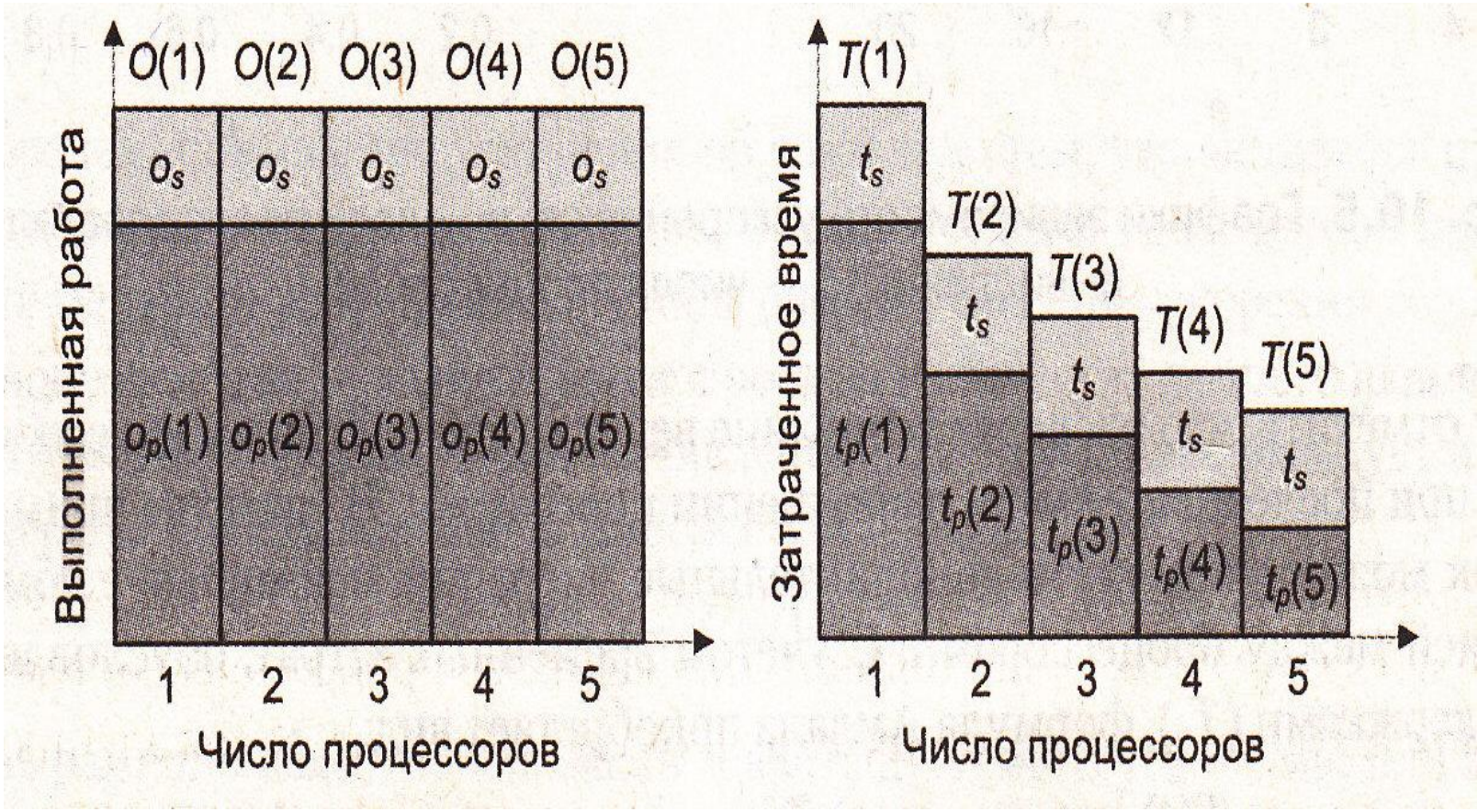
$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{fT(1) + \frac{(1-f)T(1)}{n}} = \frac{1}{f + \frac{1-f}{n}}$$

При безграничном увеличении числа процессоров формула имеет вид:

$$\lim_{n \to \infty} S = \frac{1}{f}$$

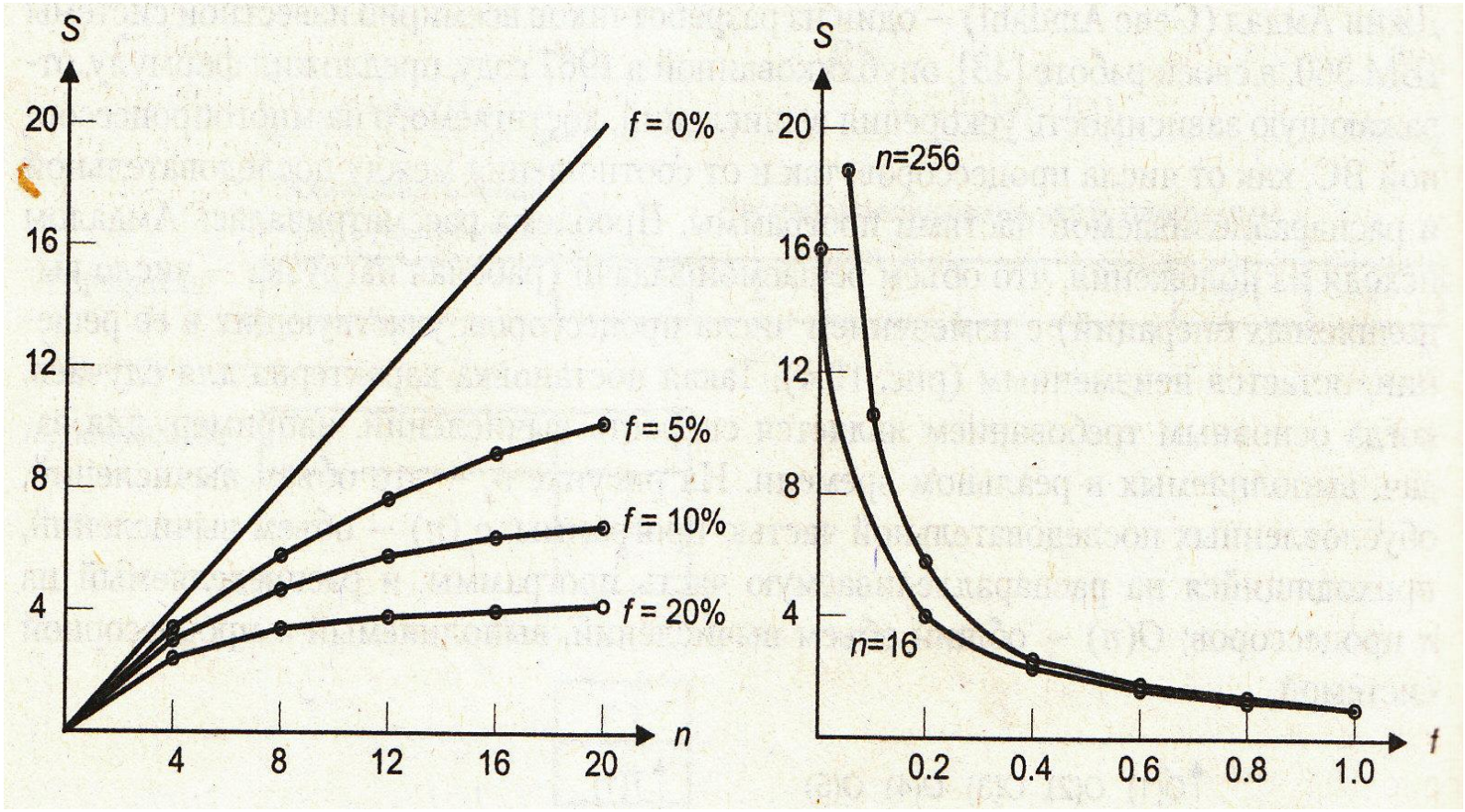
Если f равно 0.25 , то ускорения больше 4 мы не можем получить при любом n

Распределение рабочей нагрузки и времени вычислений по Амдалу



ЦЕЛЬ – ускорение вычислений при неизменном объеме задачи.

Графики закона Амдала



Необходимо также учитывать издержки, связанные с операциями обмена между процессорами.

Закон Джона Густавсона - Барсиса

NASA

Обычно, получая в свое распоряжение более мощную систему, пользователь не стремится сократить время вычислений. Он старается увеличить объем решаемой задачи, например повышая точность вычислений. Но повышение точности влечет уменьшение доли f.

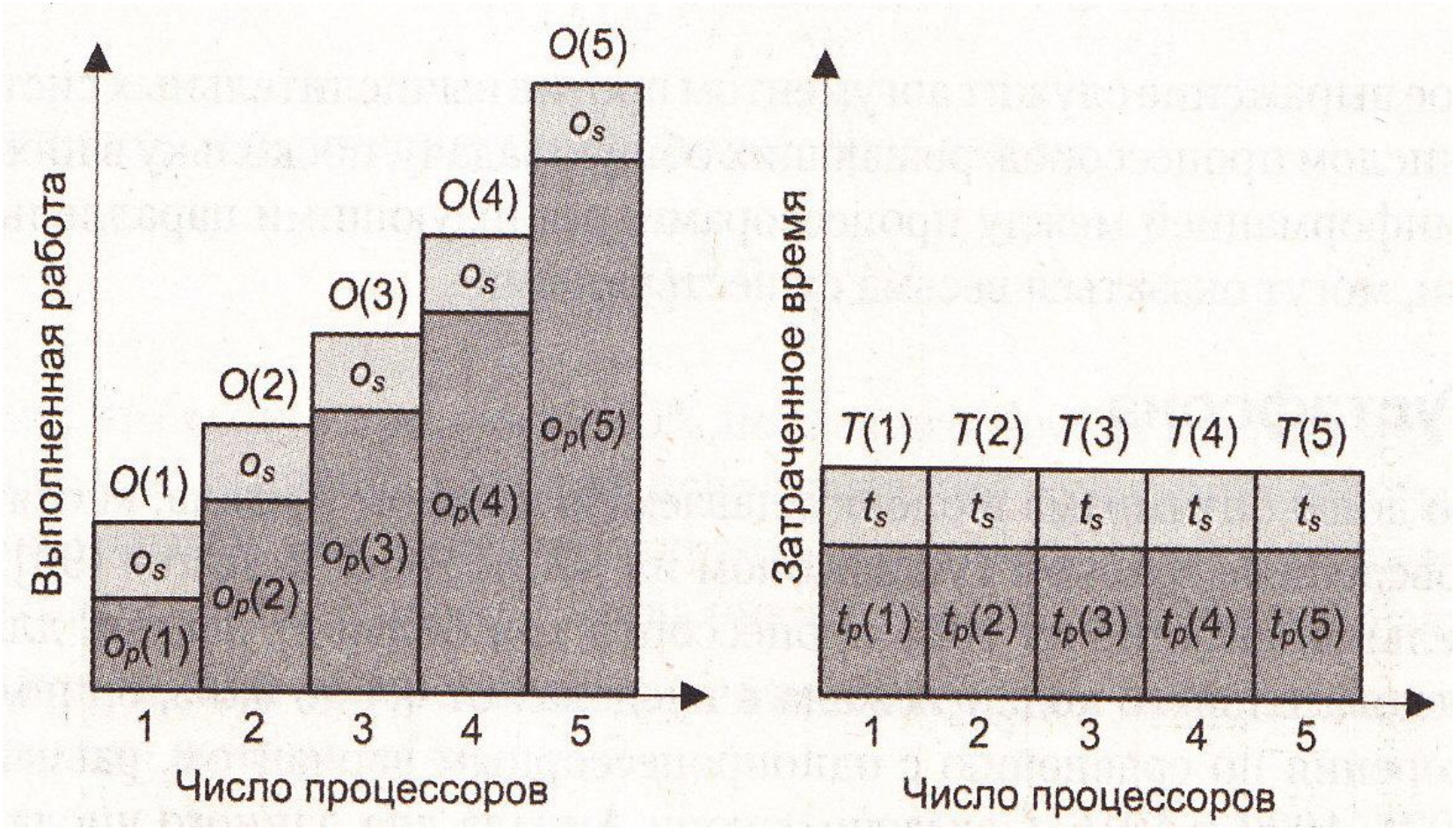
Для оценки ускорения вычислений, когда объем последних увеличивается с увеличением количества процессоров, но общее время вычислений остается постоянным

$$S(n) = f + (1 - f)n$$

Закон масштабируемого ускорения

Форма использования дополнительной мощности, возникающей при увеличении n

Распределение рабочей нагрузки и времени вычислений по Густавсону



Закон Сана-Ная – закон ускорения ограниченного памятью

Ксиан-Хе Сан и Лайонел Най

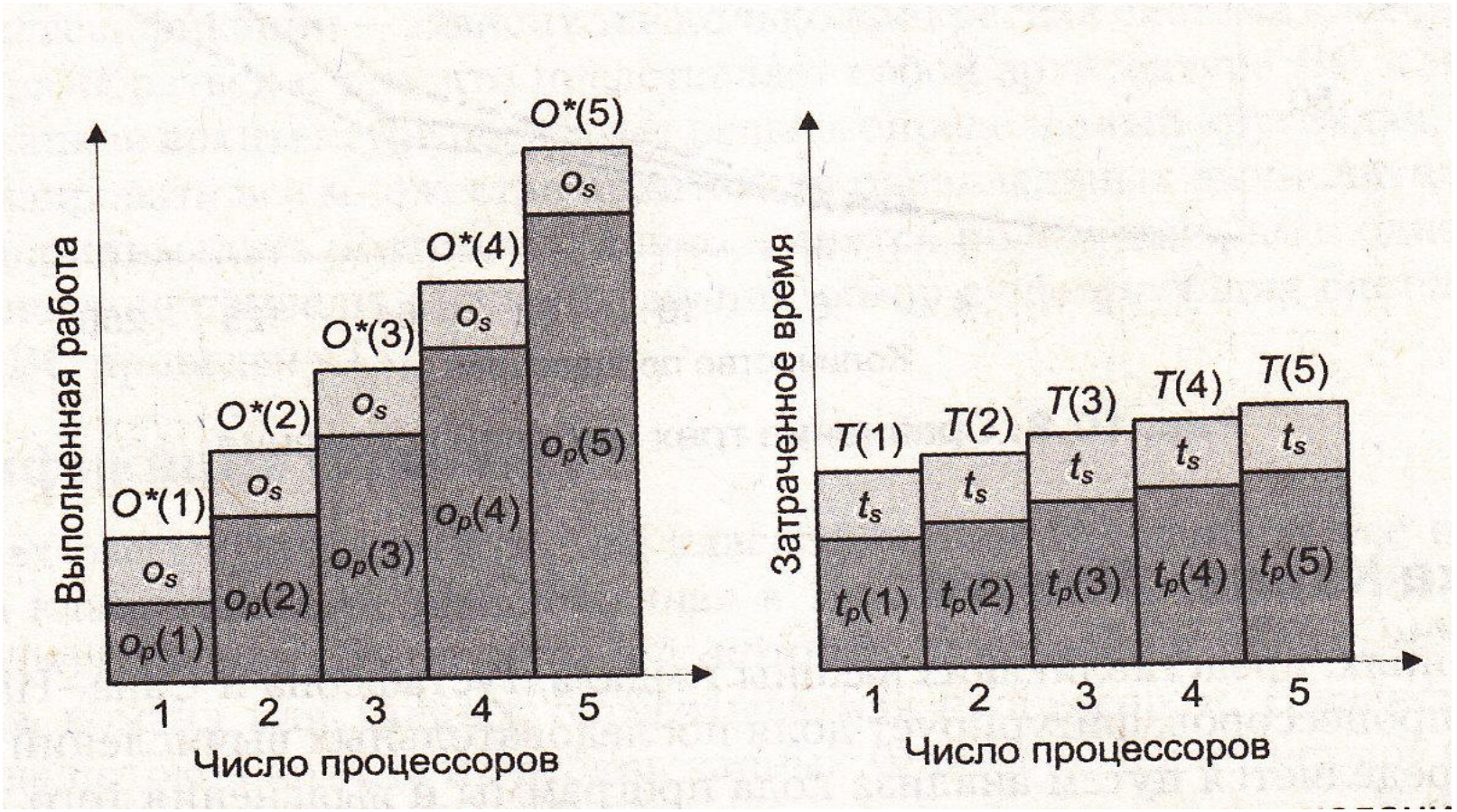
Каждый процессор имеет свою память определенного размера. При увеличении в системе количества процессоров – увеличивается общая память системы. Размер решаемой задачи ограничивается размером общей памяти.

Вводится понятие коэффициента масштабируемости распараллеливаемой части задачи G(n)

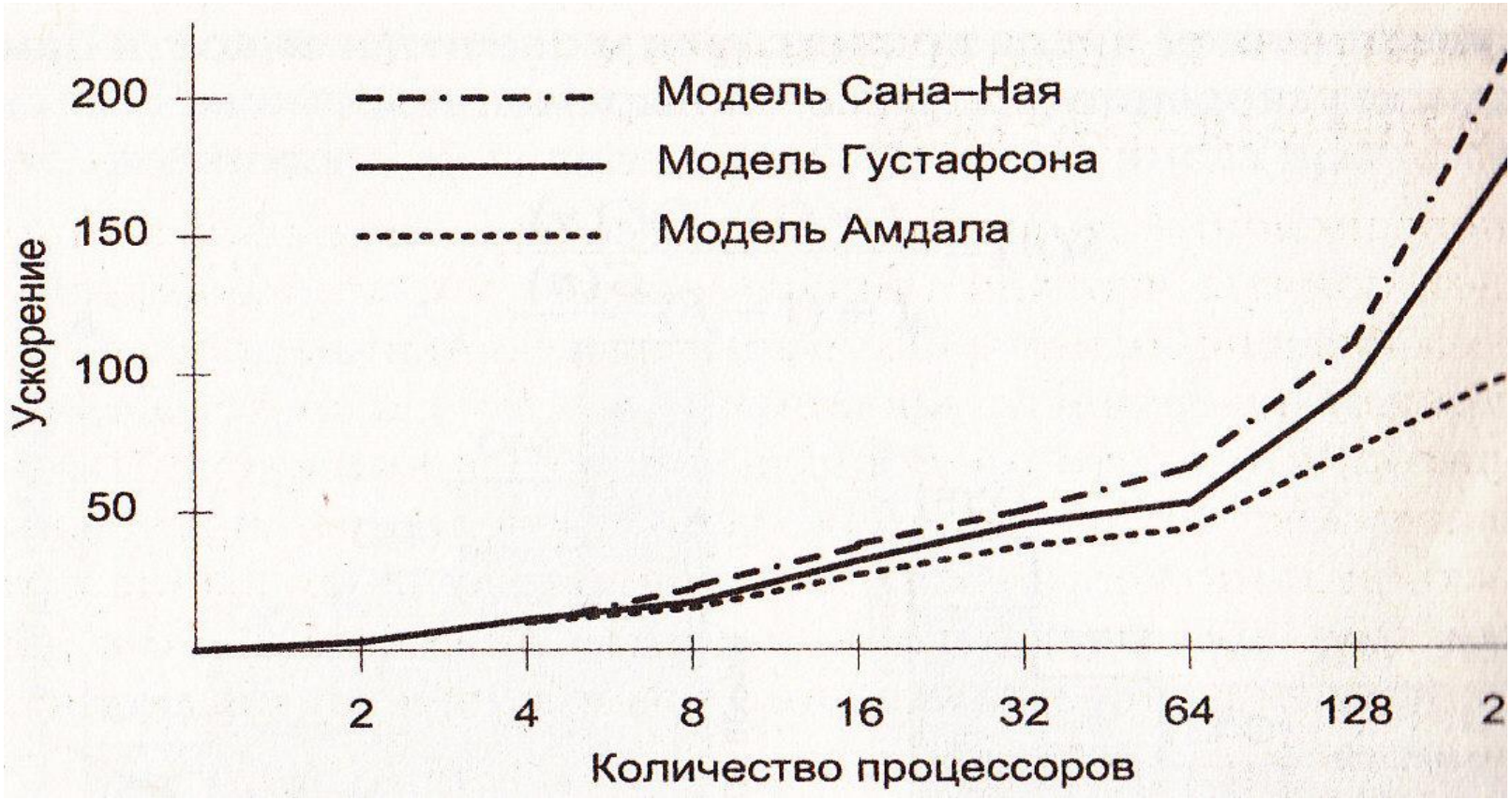
$$S(n) = \frac{f + (1 - f)G(n)}{f + (1 - f)\dfrac{G(n)}{n}}$$

При G(n)=1 приходим к закону Амдала. При G(n) = n , к закону Густавсона

Распределение рабочей нагрузки и времени вычислений по Сана и Наю



Сравнение трех моделей ускорения



Метрика Карпа-Флэтта

- Практика показывает, что значения ускорения в реальных системах ниже, чем предсказывают формулы. Так как формулы не учитывают издержек на взаимодействие процессоров.

$$e = \frac{\left(\frac{1}{S(n)}\right) - \frac{1}{n}}{1 - \frac{1}{n}}$$

Чем меньше «e» , тем лучше может быть распараллелен код. Значение ускорения получают на реальной ВС.

Анализ масштабируемости параллельных вычислений

Оценим накладные расходы возникающие при параллельных вычислениях. Они могут быть оценены следующей формулой.

$$T_0 = T_p \times P - T_1$$

Накладные расходы появляются за счет необходимости взаимодействия процессоров, например синхронизации и обмена сообщениями.

Анализ масштабируемости параллельных вычислений

Учитывая издержки можно получить формулу ускорения

$$S_p = \frac{T_1}{T_p} = \frac{PT_1}{T_1 + T_0}$$

Эффективность получим по формуле

$$E_P = \frac{S_P}{P} = \frac{T_1}{T_1 + T_0} = \frac{1}{1 + T_0 / T_1}$$

Если T1=const, то при увеличении p, To будет расти и эффективность будет падать из-за увеличения накладных расходов.