

# Моделирование и анализ параллельных вычислений

*Основные показатели  
параллельных вычислений.  
Топология методов в  
коммуникабельной среде.*

# Введение

Параллельные вычисления представляют собой область информатики, которая занимается разработкой алгоритмов и программ, выполняемых одновременно на нескольких процессорах или компьютерах для ускорения обработки данных. В этой лекции мы рассмотрим основные показатели, используемые для анализа и оценки эффективности параллельных вычислений, включая скорость, эффективность, стоимость и масштабируемость. Также будет затронута топология методов в коммуникабельной среде.

# Показатели эффективности параллельного алгоритма

1 **Ускорение**  
*(speedup)*

2 **Эффективность**  
*(Efficiency)*

3 **Стоимость**  
*(Cost)*

|       |   |
|-------|---|
| $p$   | количество процессоров, задействованных в выполнении алгоритма. |
| $T_p$ | время выполнения параллельного алгоритма.                       |
| $T_1$ | время выполнения аналогичного последовательного алгоритма       |
| $n$   | количество входных данных                                       |
| $S_p$ | величина ускорения  |
| $E_p$ | величина эффективности  |
| $C_p$ | величина стоимости вычислений                                   |



# Ускорение (Speedup)

1

## Определение

Ускорение  $S_p(n)$  - это мера того, насколько быстрее параллельный алгоритм решает задачу по сравнению с лучшим последовательным алгоритмом. Определяется как отношение времени выполнения последовательного алгоритма к времени выполнения параллельного алгоритма при использовании  $p$  процессоров.

2

## Формула

$$S_p(n) = T_1(n) / T_p(n)$$

где  $T_1(n)$  - время выполнения последовательного алгоритма, а  $T_p(n)$  - время выполнения параллельного алгоритма при использовании  $p$  процессоров.

3

## Оценка максимально достижимого параллелизма

Получение идеальных величин  $S_p=p$  для ускорения и  $E_p=1$  для эффективности может быть обеспечено не для всех вычислительно трудоемких задач.

# Эффективность (Efficiency)

1

## Определение

определяется как отношение ускорения к числу используемых процессоров и показывает среднюю долю времени, в течение которого каждый процессор активно участвует в вычислении

2

## Формула

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

3

## Примечание

(величина эффективности определяет среднюю долю времени выполнения параллельного алгоритма, в течение которого процессоры реально используются для решения задачи)

# Стоимость (Cost)

1

## Определение

Стоимость вычислений  $C_p$  представляет собой произведение количества процессоров на время выполнения параллельного алгоритма

2

## Формула

$$C_p = p \cdot T_p(n)$$

3

## Примечание

*Показатели качества параллельных вычислений являются противоречивыми: попытки повышения качества параллельных вычислений по одному из показателей (ускорению или эффективности) может привести к ухудшению ситуации по другому показателю*

# Закон Амдаля

## 1 Сущность закона

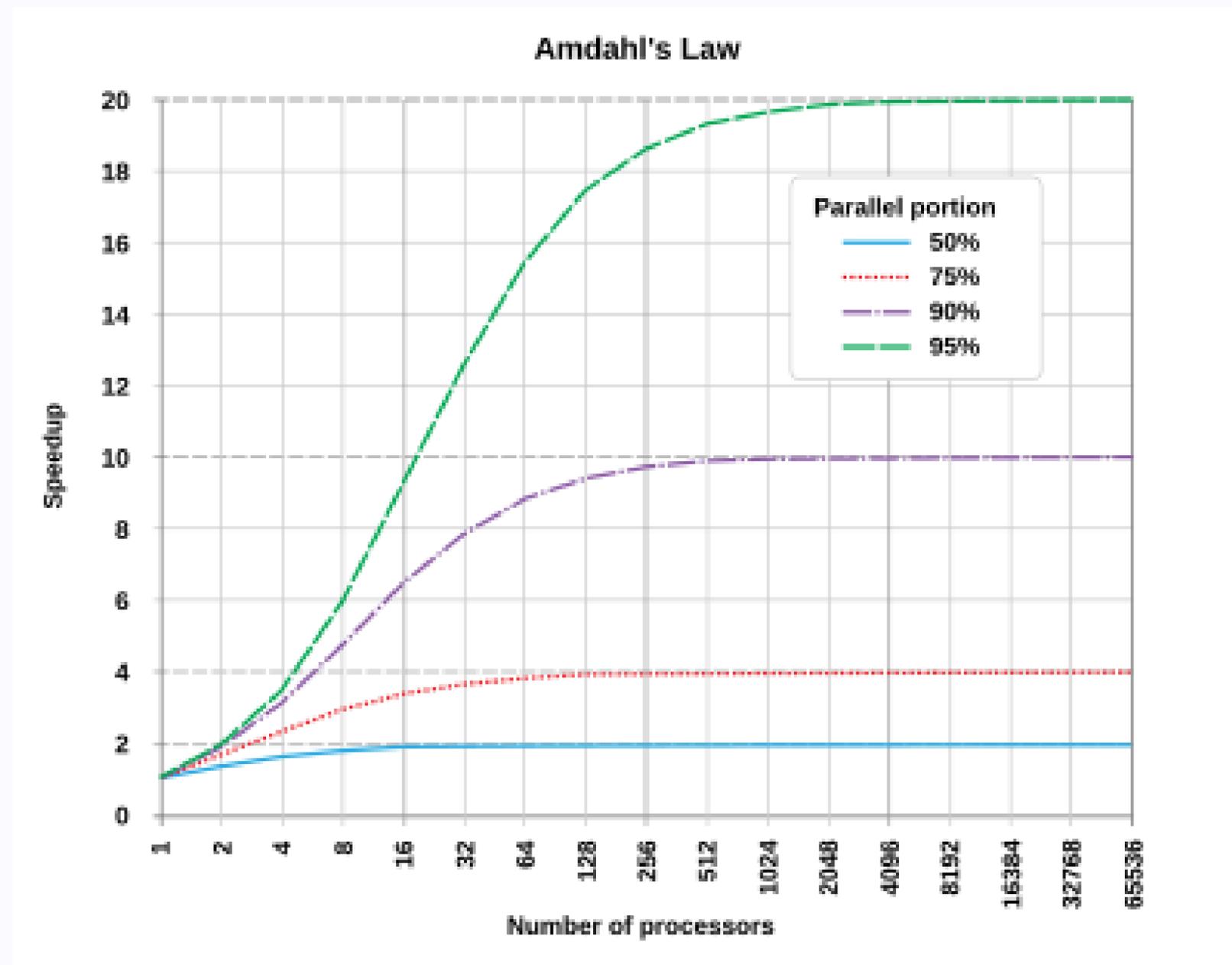
Закон Амдаля демонстрирует, что ускорение ограничено из-за наличия последовательных частей алгоритма. Если  $f$  - доля времени, необходимого для выполнения последовательных операций, то максимальное ускорение  $S_p$  будет ограничено.

## 2 Формула

$$S_p \leq \frac{1}{f + \frac{1-f}{p}}$$

## 3 Вывод

Закон Амдаля показывает, что для достижения значительного ускорения необходимо минимизировать долю последовательных операций в алгоритме.



# Закон Густавсона – Барсиса

## 1 Сущность закона

оценка максимально достижимого ускорения выполнения параллельной программы, в зависимости от количества одновременно выполняемых потоков вычислений («процессоров») и доли последовательных расчётов. Закон Густавсона-Барсиса предлагает альтернативный взгляд на масштабирование ускорения, учитывая изменение размера задачи при увеличении количества процессоров

## 2 Формула

$$S_p = g + (1 - g) \times p$$

где  $g$  - доля времени, затрачиваемого на последовательные вычисления.

## 3 Вывод

Оценку ускорения, получаемую в соответствии с законом Густавсона-Барсиса, еще называют ускорением масштабирования (scaled speedup), поскольку данная характеристика может показать, насколько эффективно могут быть организованы параллельные вычисления при увеличении сложности решаемых задач.

# Анализ Масштабируемости

Параллельный алгоритм считается масштабируемым, если увеличение числа процессоров приводит к увеличению ускорения при сохранении эффективности.

## Основные формулы:

- Общие накладные расходы

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

- Время параллельного решения:

$$T_p = \frac{T_1 + T_0}{p}$$

- Ускорение

$$S_p = \frac{p \times T_1}{T_1 + T_0}$$

- Эффективность:

$$E_p = \frac{T_1}{T_1 + T_0}$$

При фиксированном числе процессоров увеличение сложности задачи улучшает эффективность, а при увеличении числа процессоров возможно поддержание эффективности за счет повышения сложности задач.

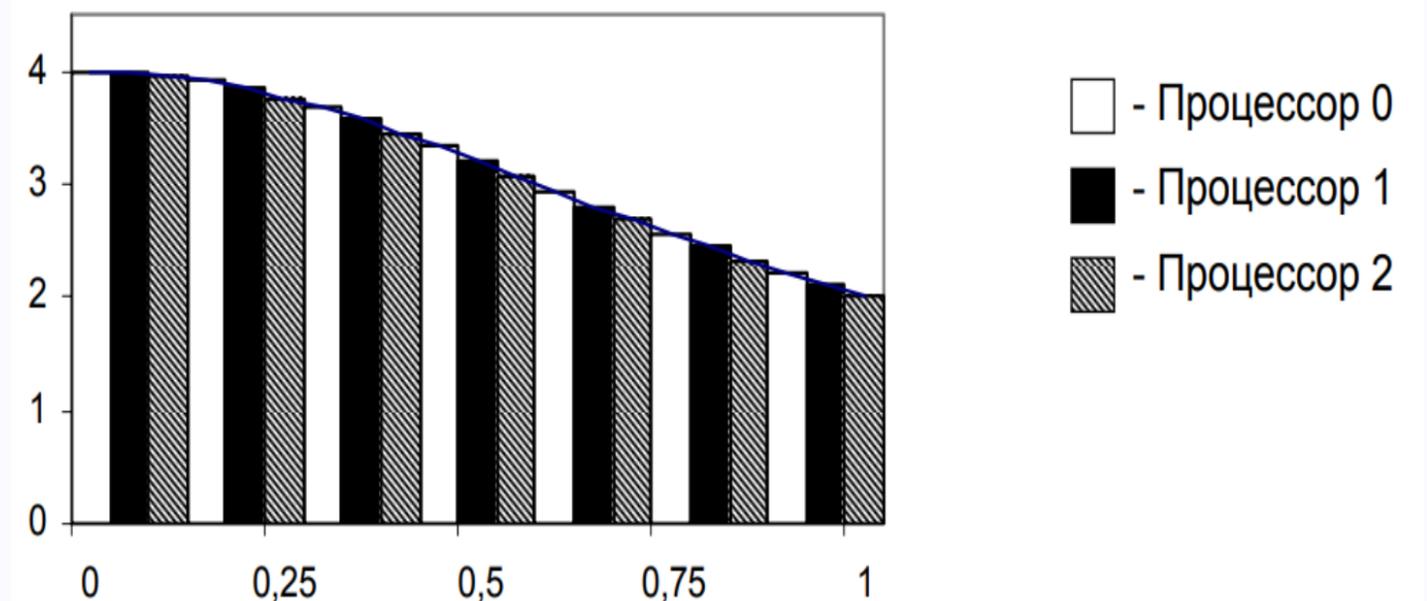
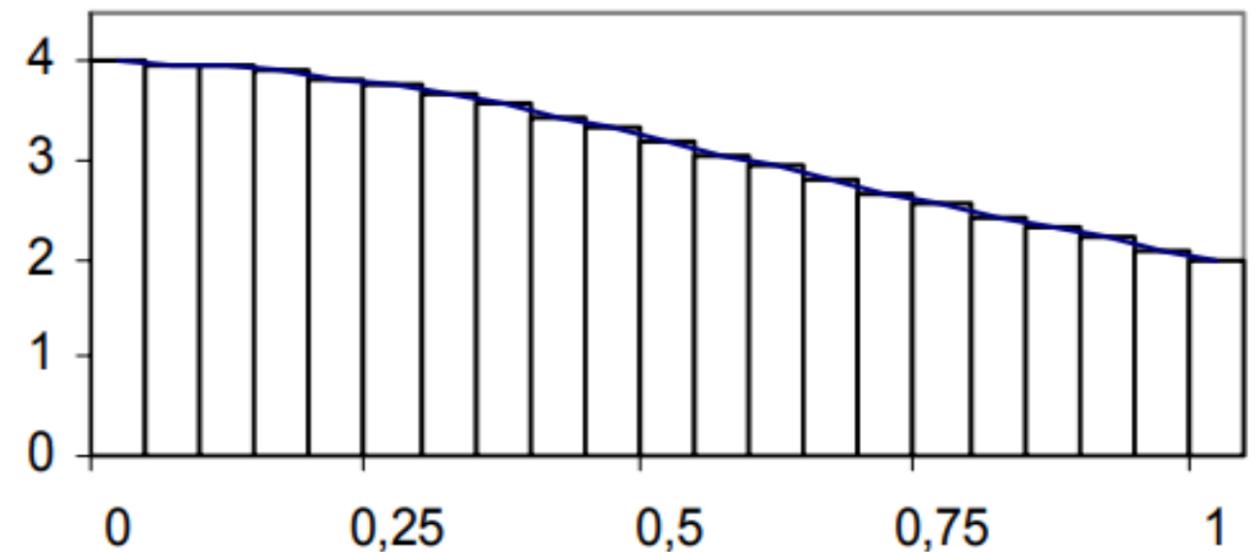
Функция изоэффективности показывает зависимость между числом процессоров и сложностью задачи для поддержания постоянной эффективности:

$$T_0 / T_1 = \frac{(1 - E)}{E}$$

# Пример: Вычисление числа $\pi$

- Значение числа  $\pi$  может быть получено при помощи интеграла.
- Для численного интегрирования применим метод прямоугольников.
- Распределим вычисления между  $p$  процессорами (циклическая схема).
- Получаемые на отдельных процессорах частные суммы должны быть просуммированы

$$\int_0^1 \frac{4}{1+x^2} dx$$



# Пример: Вычисление числа $\pi$

## Анализ эффективности

- $n$  – количество разбиений отрезка  $[0,1]$
- Вычислительная сложность задачи

$$W = T_1 = 6n$$

- Количество узлов сетки на отдельном процессоре

$$m = \lceil n/p \rceil \leq n/p + 1$$

- Объем вычислений на отдельном процессоре

$$W_p = 6m = 6n/p + 6.$$

- Время параллельного решения задачи

$$T_p = 6n/p + 6 + \log_2 p$$

- Ускорение

$$S_p = T_1 / T_p = 6n / (6n/p + 6 + \log_2 p)$$

- Эффективность

$$E_p = 6n / (6n + 6p + p \log_2 p)$$

- Функция изоэффективности

$$W = K(pT_p - W) = K(6p + p \log_2 p) \\ \Rightarrow n = [K(6p + p \log_2 p)]/6, \quad (K = E/(1-E))$$

- Пример:  $E=0.5$

при  $p=8 \rightarrow n=12$

при  $p=64 \rightarrow n=128$

# Топология методов в коммуникабельной среде



## Линейная

Каждый процессор связан с двумя соседями.



## Дерево

Иерархическая структура с родительскими и дочерними узлами.



## Сетка

Процессоры размещены в форме матрицы, обеспечивая связь по вертикали и горизонтали.



## Кольцевая

Процессоры образуют кольцо.

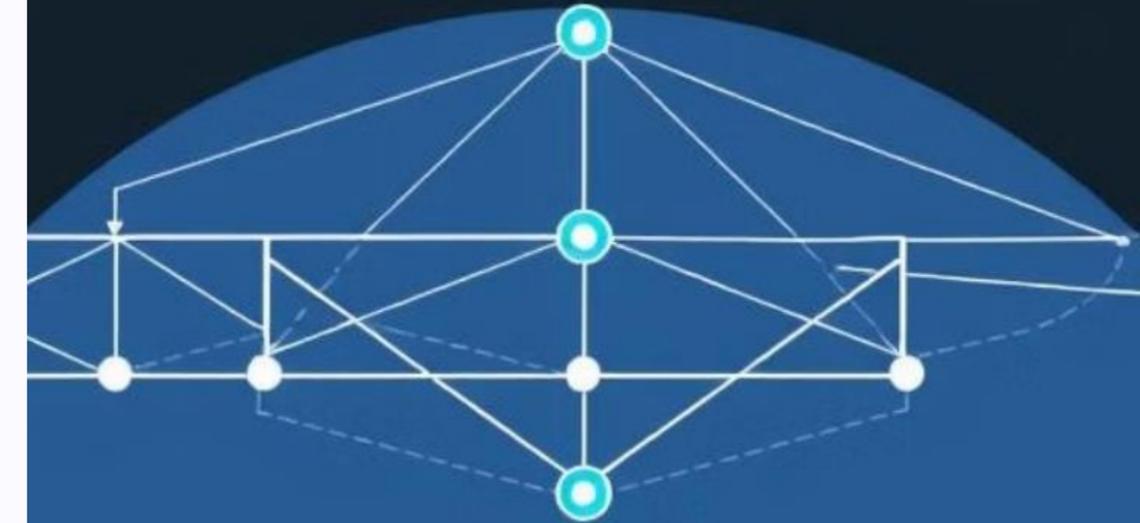


## Гиперкуб

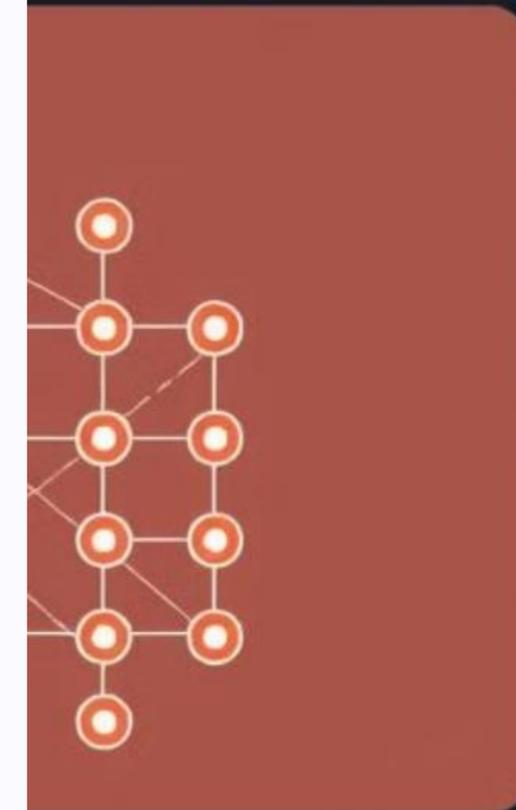
Каждый процессор связан с  $\log_2 n$  других процессорами.

*Эффективность коммуникаций влияет на общую производительность системы, и выбор топологии зависит от характера задач и алгоритмов.*

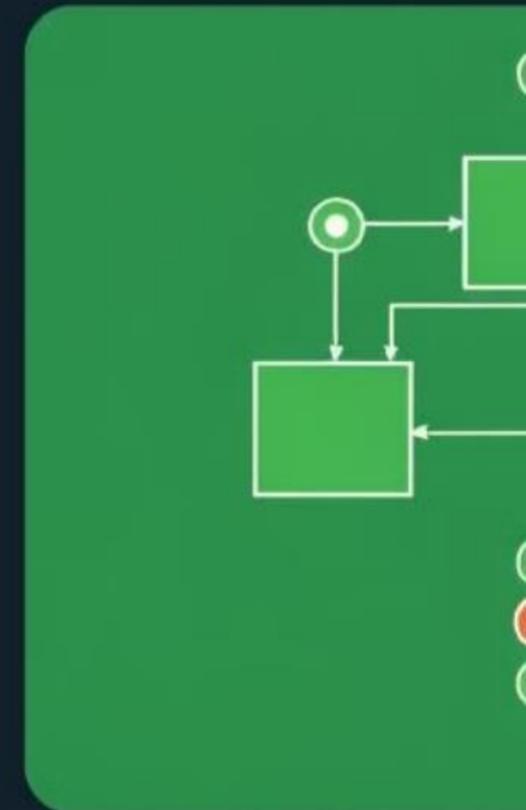
elem couwe'r tnon



structure



Grid str



# Заключение

## Значение параллельных вычислений

Параллельные вычисления оставляют значительный след в разработке алгоритмов, оптимизации ресурсов и ускорении обработки данных.

## Важность понимания показателей

Понимание ускорения, эффективности, стоимости и масштабируемости критически важно для создания эффективных вычислительных систем.

## Законы и модели

Законы Амдаля и Густавсона-Барсиса помогают нам оценить возможности и ограничения в распараллеливании задач.



**Благодарю за внимание!**