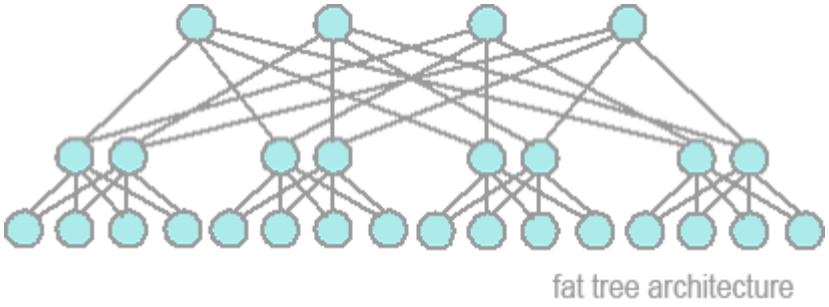
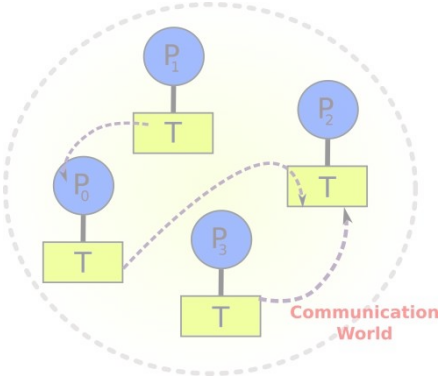
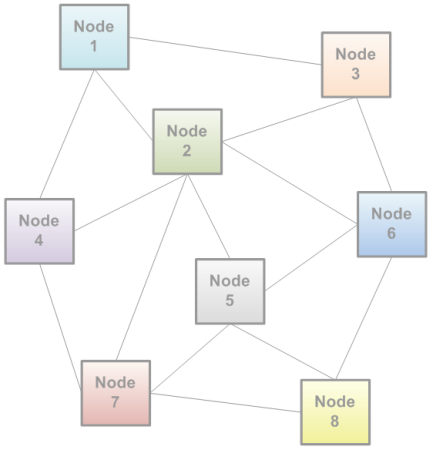
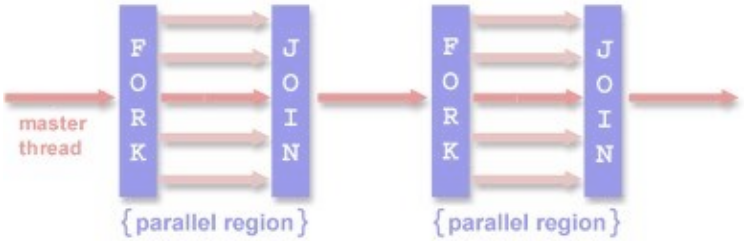




Введение в параллельное программирование



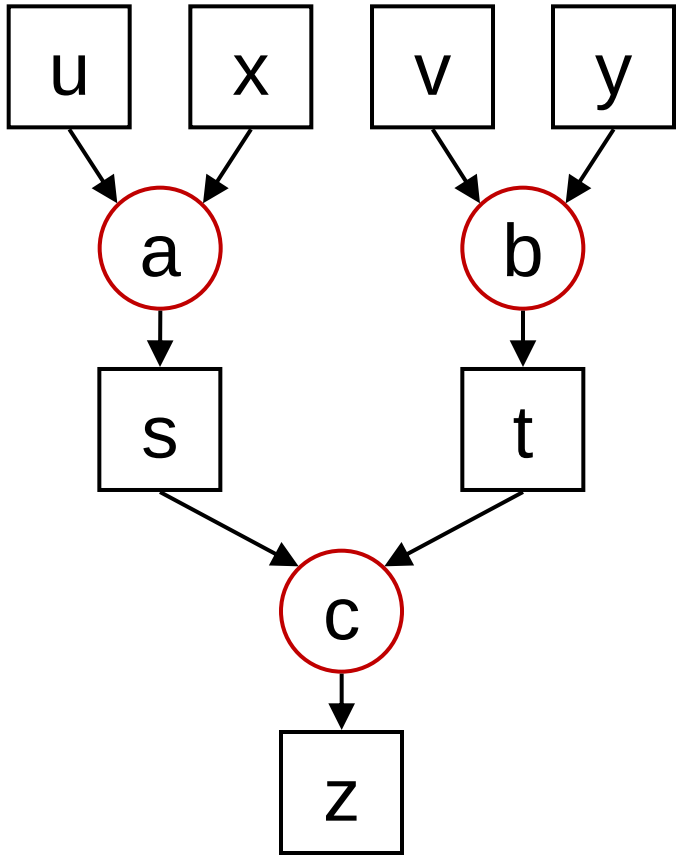


От алгоритма к программе
Задача: z = u·x + v·y

- **Программа**
  1. load r1, u
  2. load r2, x
  3. mul r3, r1,r2
  4. load r1, v
  5. load r2, y
  6. mul r4, r1,r2
  7. add r1, r3,r4
  8. store z, r1

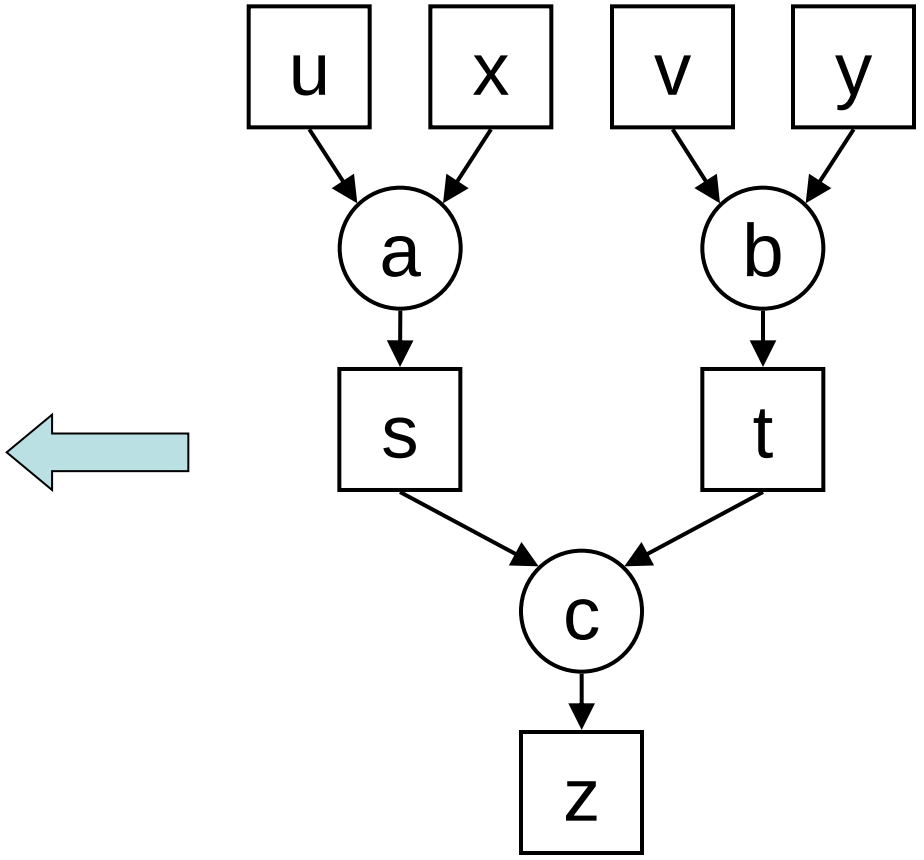
От алгоритма к программе
Задача: z = u·x + v·y

- **Программа**
  1. load r1, u
  2. load r2, x
  3. mul r3, r1,r2
  4. load r1, v
  5. load r2, y
  6. mul r4, r1,r2
  7. add r1, r3,r4
  8. store z, r1



От алгоритма к программе
Задача: z = u·x + v·y

- **Программа**
  1. load r1, u
  2. load r2, x
  3. mul r3, r1,r2
  4. load r1, v
  5. load r2, y
  6. mul r4, r1,r2
  7. add r1, r3,r4
  8. store z, r1



- Упорядочили операции
- Распределили ресурсы

Основные определения

- **Алгоритм**

  рекурсивно-перечислимое множество функциональных термов

  (соответствует определению вычислимой функции по Клини)

  A = (X, F, in, out)

  X = { x,y,…,z } – множество переменных

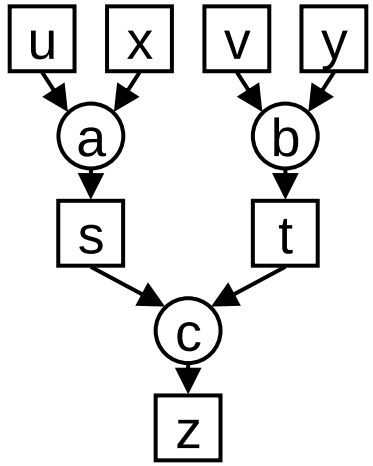
  (единственного присваивания)

  F = { a,b, …,c } – множество операций

  (однократного срабатывания)

  in: F → X – отношение «вход»

  out: F → X – отношение «выход»



Алгоритм удобно рисовать в виде ориентированного графа с вершинами-переменными и вершинами-операциями.

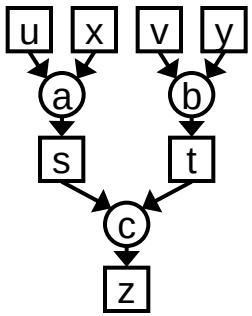
Основные определения

- Алгоритм: A = (X, F, in, out)
- **Представление алгоритма**

  P = (X, F, in, out, C, M)

  C: F → F – управление:
  - множество ограничений на порядок исполнения операций,
  - включает минимальное (потоковое) управление.

  M – отображение переменных и операций на ресурсы вычислителя.



X = { x,y,z,u,v,s,t }
F = { a,b,c }
in = { (a,u), (a,x), (b,u), (b,y), (c,s), (c,t) }
out = { (a,s), (a,t), (c,z) }

C = { (a,c), (b,c) }
M = { }

load r1, u	X = { x,y,z,u,v,s,t }
load r2, x	F = { a,b,c }
mul r3, r1,r2	in = { (a,u), (a,x), (b,u), (b,y), (c,s), (c,t) }
load r1, v	out = { (a,s), (a,t), (c,z) }
load r2, y	
mul r4, r1,r2	C = { (a,c), (b,c), (a, b) }
add r1, r3,r4	M = { (u,r1), (x,r2), (s,r3),
store z, r1	(v,r1), (y,r2), (t,r4), (z,r1),
	(a, core), (b, core), (c, core) }

Основные определения

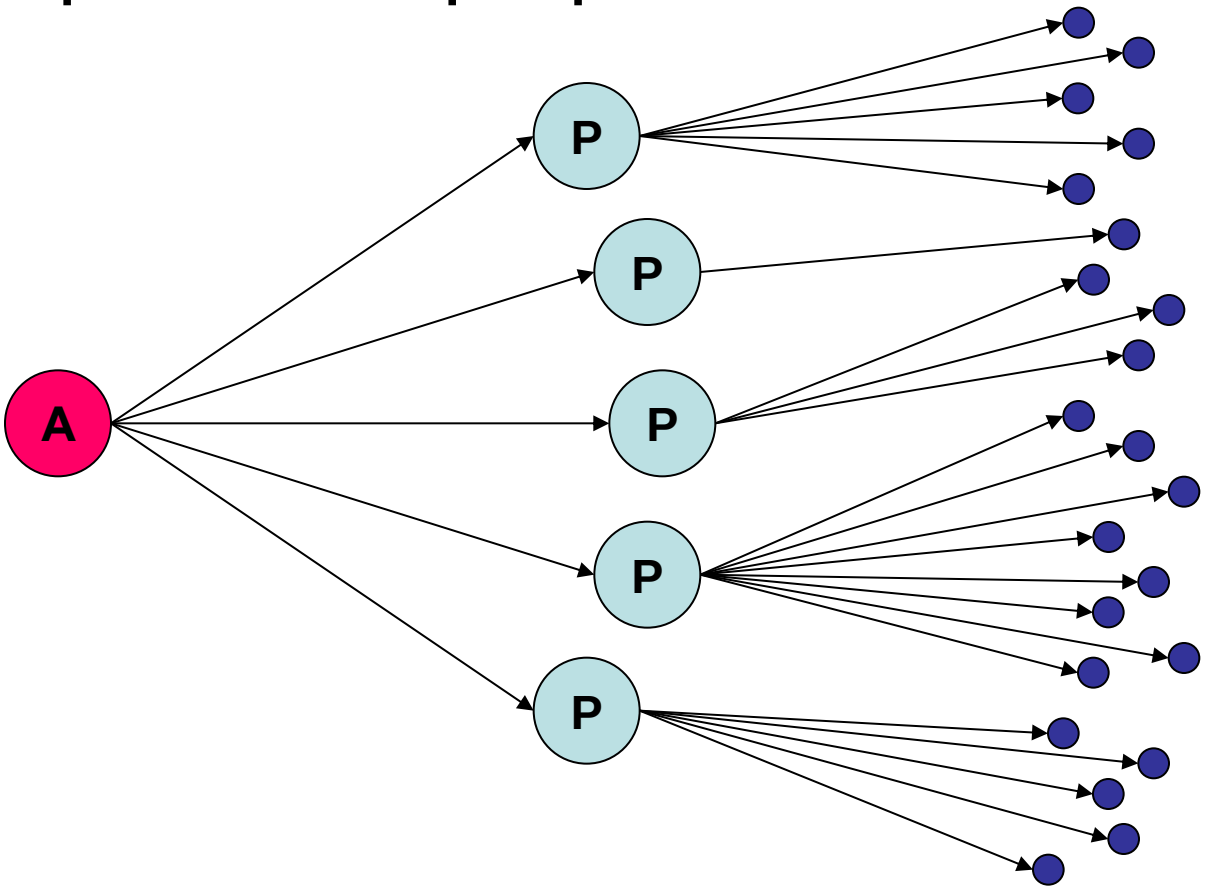
- Алгоритм: A = (X, F, in, out)
- Представление алгоритма: P = (X,F,in,out,C,M)
- **Программа** – такое представление алгоритма, которое может быть исполнено вычислителем.

Основные определения

- Алгоритм: A = (X, F, in, out)
- Представление алгоритма: P = (X,F,in,out,C,M)
- **Программа** – такое представление алгоритма, которое может быть исполнено вычислителем.
- **Реализация** алгоритма A в представлении P (реализация программы P) – это процесс выполнения операций алгоритма A в порядке, который не противоречит управлению C.

Основные определения

Алгоритм → Программа → Реализация



Основные определения

- Пример: последовательная программа

  C – управление
  - Последовательное управление

  M – отображение переменных на ресурсы
  - Программист: отображение данных на статическую память, стек, кучу
  - Компилятор: отображение данных на регистры
  - Ядро процессора: переименование регистров, отображение данных в кэш

  M – отображение операций на ресурсы
  - Компилятор: переупорядочение операций
  - Операционная система: выбор ядра процессора
  - Ядро процессора: суперскалярное исполнение, переупорядочение операций

Основные определения

- Пример: параллельная программа

  C – управление
  - Задаётся программистом

  M – отображение переменных на ресурсы
  - …
  - …
  - …
  - Распределение данных по узлам, по уровням иерархии памяти, по памяти сопроцессоров

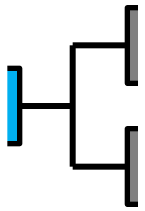
  M – отображение операций на ресурсы
  - …
  - …
  - …
  - Распределение операций по узлам, ядрам, сопроцессорам

Модели параллельных вычислителей

- Основные компоненты вычислителей
  - Исполнительные устройства

    - исполнительные устройства, ядра, процессоры, вычислительные узлы, кластера, …
  - Блоки памяти

    - Оперативная память, кэш-память, дисковая память …
  - Сети связи
    - Процессор-память, межпроцессорные, межузловые, межкластерные, …

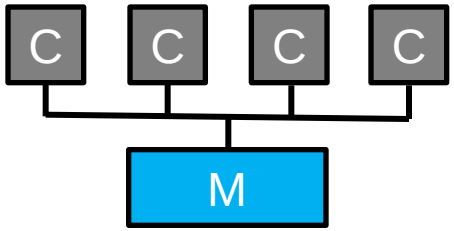


Модели параллельных вычислителей

- Основные классы вычислителей

Системы с общей памятью

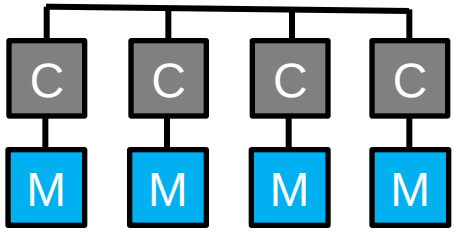
(мультипроцессоры)



- Многоядерные процессоры
- Многопроцессорные узлы
- …

Системы с распределенной памятью

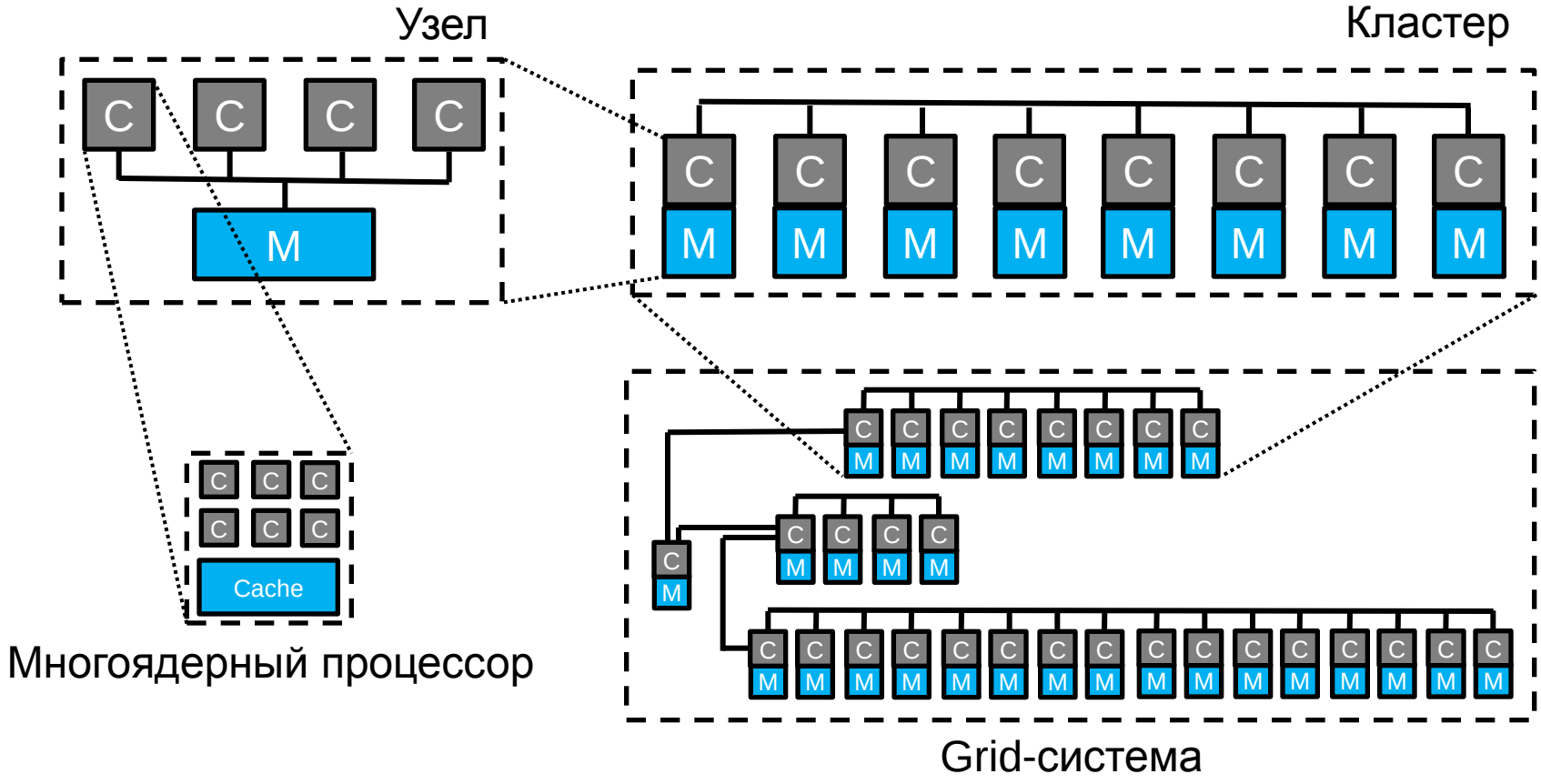
(мультикомпьютеры)



- Сети рабочих станций
- Кластера
- Grid
- …

Модели параллельных вычислителей

- Иерархия вычислителей



Меры качества параллельных программ

- **Время работы**
  - Сколько времени программа работает на N ядрах?
- **Ускорение**
  - Во сколько раз программа стала быстрее работать на N ядрах по сравнению с одним ядром / последовательной программой?
- **Эффективность распараллеливания**
  - Какой процент времени работы программы идёт на полезную работу?
- **Масштабируемость**
  - Как быстро эффективность падает с ростом числа ядер?
- **…**

Меры качества параллельных программ

- **Ускорение**

  Ускорение параллельной программы при использовании N исполнительных устройств относительно…

  - последовательной:

    $$S^{seq}_N = T_{seq} / T_N$$

  - параллельной с использованием одного исполнительного устройства:

    $$S^1_N = T_1 / T_N$$

где:

T_{seq} – время работы последовательной программы,

T_1 – время работы параллельной программы при использовании одного исполнительного устройства,

T_N – время работы параллельной программы при использовании N исполнительных устройств.

Меры качества параллельных программ

- **Эффективность** распараллеливания

  Эффективность использования N исполнительных устройств относительно…

  - последовательной программы:

    $$E^{seq}_N = S^{seq}_N / N$$

  - параллельной программы с использованием одного исполнительного устройства:

    $$E^{1}_N = S^{1}_N / N$$

где:

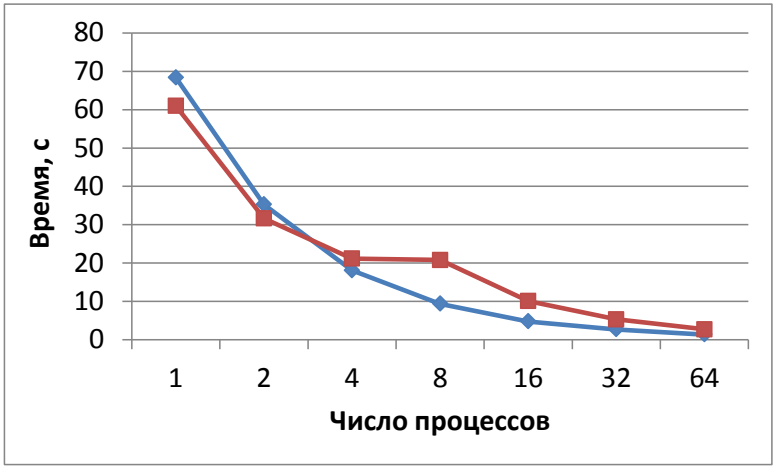
S^{seq}_N – ускорение параллельной программы при использовании N исполнительных устройств относительно последовательной,

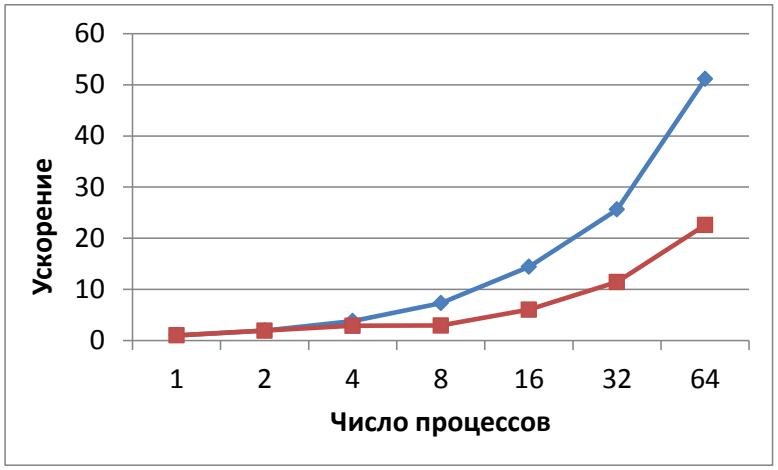
S^{1}_N – ускорение параллельной программы при использовании N исполнительных устройств относительно параллельной при использовании одного исполнительного устройства.

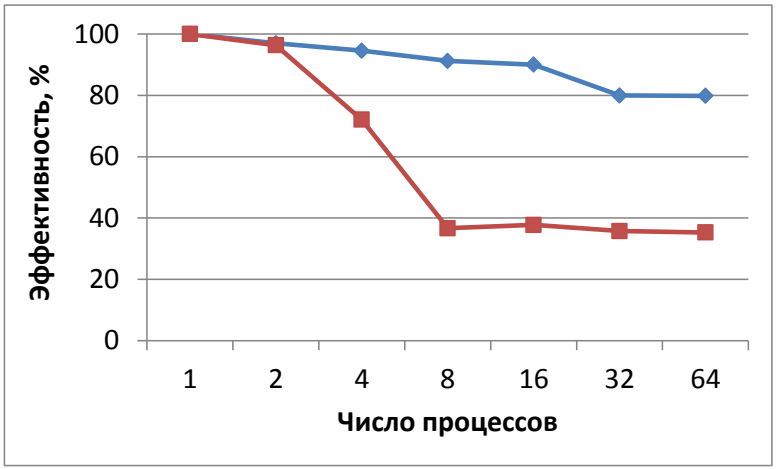
Меры качества параллельных программ

Пример:

- Программа хорошо масштабируется
- Программа плохо масштабируется







Предел ускорения: закон Амдала

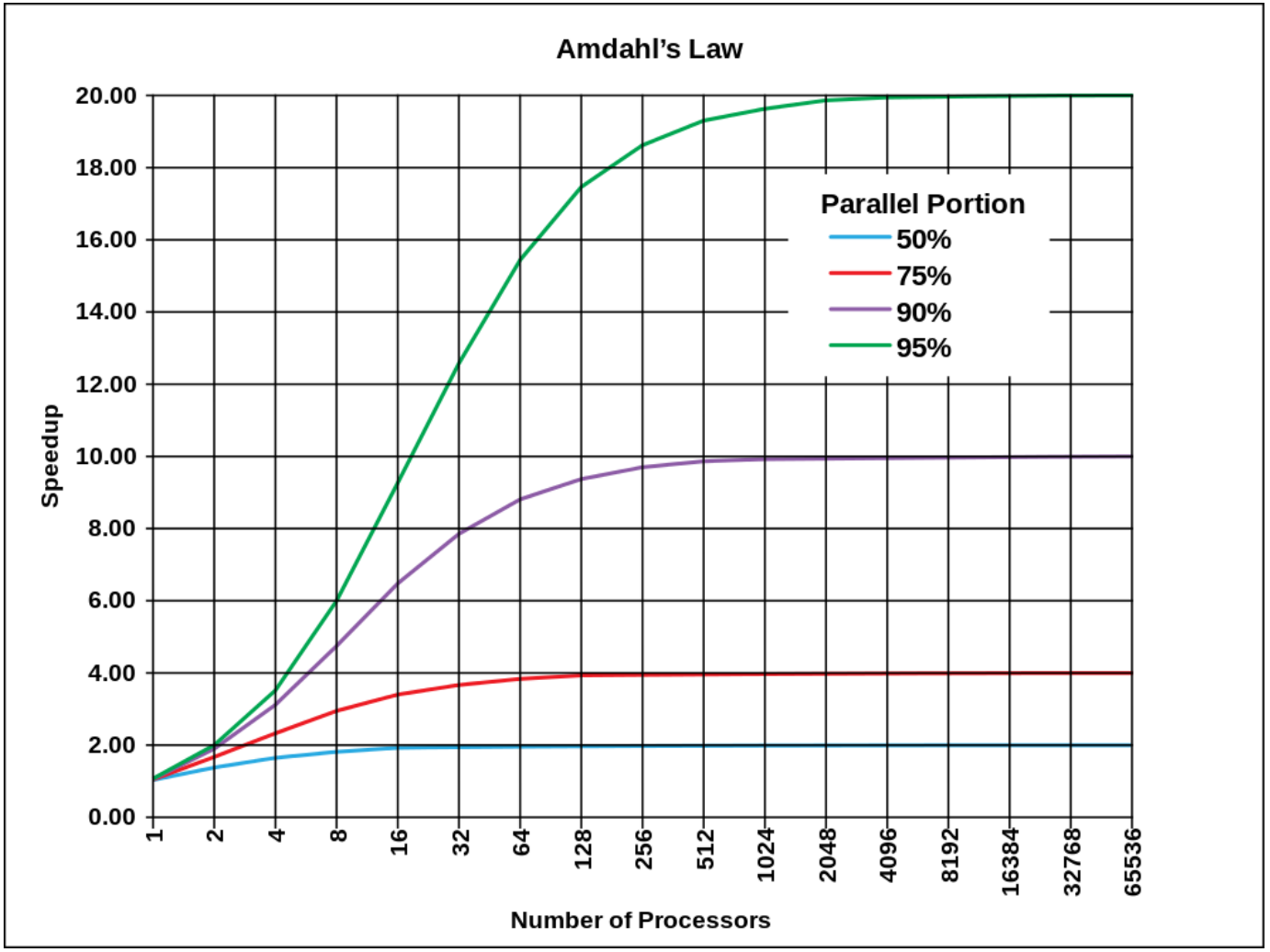
- Максимальное ускорение, которое можно получить при использовании N исполнительных устройств:

$$S_N^1 = \frac{1}{(1-P) + \frac{P}{N}}$$

P – доля вычислений, которые могут быть выполнены параллельно,

(1–P) – доля последовательных вычислений.

Предел ускорения: закон Амдала



Wikipedia