

# Высокопроизводительные вычисления

## 1.2. Базовые методы ускорения вычислений основные понятия распараллеливания

Негода В.Н., д.т.н., профессор  
кафедры ВТ УлГТУ

# Факторы, определяющие время исполнения программных функций

- Свойства аппаратно-программной платформы
- Возможности программиста в полной мере использовать свойства аппаратно-программной платформы: знание свойств, умение ими пользоваться в целях ускорения программ,
- Быстродействие используемых алгоритмов
- Функциональные возможности инструментальных средств программирования: оптимизаторы, профилировщики, высокоскоростные библиотеки функций
- Свойства рабочей нагрузки, т.е. данных, вовлекаемых в обработку
- Контекст процесса исполнения: характер фоновой нагрузки; механизмы диспетчеризации процессов, объемы данных, сохраняемых и восстанавливаемых при переключении задач
- Требования к точности обработки данных

# Наиболее важные свойства аппаратно-программных платформ, влияющие на быстродействие программ

- Параметры различных видов памяти и механизмы их использования: кэш-память различных уровней, основная память, дисковая память, механизмы виртуализации памяти
- Различные виды параллельности и средства их использования
- Инерционность системы ввода-вывода
- Инерционность системных вызовов ОС
- Накладные расходы на организацию параллельной обработки данных
- Свойства средств поддержки измерения затрат времени
- Свойства средств поддержки режима реального времени и режима разделения времени

# Два способа ускорения быстродействия машин

## 1. Повышение частоты + Увеличение памяти:

Увеличение частоты в  $k$  раз  $k * F \Rightarrow T/k$

## 2. Совершенствование архитектуры

На одном из первых компьютеров мира - EDSAC, появившемся в 1949 году в Кембридже и имевшем время такта 2 микросекунды, можно было выполнить  $2 * n$  арифметических операций за  $18 * n$  миллисекунд, то есть в среднем 100 арифметических операций в секунду.

Современный персональный компьютер способен выполнять более 100 GFLOPS при частоте 3 ГГц (такт равен 0,33 нс).

За 67 лет производительность выросла более, чем в миллиард раз. При этом выигрыш в быстродействии, связанный с уменьшением времени такта с 2 микросекунд до 0,33 наносекунды, составляет лишь около 6000 раз.

Архитектурное совершенствование дает прирост быстродействия в  $1000000000/6000 = 166667$  раз больше, нежели увеличение частоты.

# История параллельности: многоразрядность

**IBM 701 (1953), IBM 704 (1955): разрядно-параллельная память, разрядно-параллельная арифметика.**

Все самые первые компьютеры (EDSAC, EDVAC, UNIVAC) имели разрядно-последовательную память, из которой слова считывались последовательно бит за битом. Первым коммерчески доступным компьютером, использующим разрядно-параллельную память (на CRT) и разрядно-параллельную арифметику, стал IBM 701, а наибольшую популярность получила модель IBM 704 (продано 150 экз.), в которой, помимо сказанного, была впервые применена память на ферритовых сердечниках и аппаратное АУ с плавающей точкой.

# История параллельности: распараллеливание ввода-вывода

## IBM 709 (1958): независимые процессоры ввода/вывода.

Процессоры первых компьютеров сами управляли вводом/выводом. Однако скорость работы самого быстрого внешнего устройства, а по тем временам это магнитная лента, была в 1000 раз меньше скорости процессора, поэтому во время операций ввода/вывода процессор фактически простаивал. В 1958г. к компьютеру IBM 704 присоединили 6 независимых процессоров ввода/вывода, которые после получения команд могли работать параллельно с основным процессором, а сам компьютер переименовали в IBM 709. Данная модель получилась удивительно удачной, так как вместе с модификациями было продано около 400 экземпляров, причем последний был выключен в 1975 году - 20 лет существования!

## История параллельности: конвейер команд

**ATLAS (1963): конвейер команд.**

Впервые конвейерный принцип выполнения команд был использован в машине ATLAS, разработанной в Манчестерском университете. Выполнение команд разбито на 4 стадии: выборка команды, вычисление адреса операнда, выборка операнда и выполнение операции.

Конвейеризация позволила уменьшить время выполнения команд с 6 мкс до 1,6 мкс. Данный компьютер оказал огромное влияние, как на архитектуру ЭВМ, так и на программное обеспечение: в нем впервые использована мультипрограммная ОС, основанная на использовании виртуальной памяти и системы прерываний.

# История параллельности: независимые функциональные устройства

## CDC 6600 (1964): независимые функциональные устройства.

Фирма Control Data Corporation (CDC) при непосредственном участии одного из ее основателей, Сеймура Р.Крэя (Seymour R.Cray) выпускает компьютер CDC-6600 - первый компьютер, в котором использовалось несколько независимых функциональных устройств. Для сравнения с сегодняшним днем приведем некоторые параметры компьютера:

- время такта 100нс,
- производительность 2-3 млн. операций в секунду,
- оперативная память разбита на 32 банка по 4096 60-ти разрядных слов,
- цикл памяти 1мкс,
- 10 независимых функциональных устройств.



# История параллельности: независимые функциональные устройства

## CDC 6600 (1964): независимые функциональные устройства.

Фирма Control Data Corporation (CDC) при непосредственном участии одного из ее основателей, Сеймура Р.Крэя (Seymour R.Cray) выпускает компьютер CDC-6600 - первый компьютер, в котором использовалось несколько независимых функциональных устройств. Для сравнения с сегодняшним днем приведем некоторые параметры компьютера:

- время такта 100нс,
- производительность 2-3 млн. операций в секунду,
- оперативная память разбита на 32 банка по 4096 60-ти разрядных слов,
- цикл памяти 1мкс,
- 10 независимых функциональных устройств.

IA-64, видео-процессоры, арифметические сопроцессоры и масса независимых контроллеров ввода-вывода

## **История параллельности: конвейерные независимые функциональные устройства**

**CDC 7600 (1969): конвейерные независимые функциональные устройства.**

CDC выпускает компьютер CDC-7600 с восемью независимыми конвейерными функциональными устройствами - сочетание параллельной и конвейерной обработки.

Основные параметры:

- такт 27,5 нс,
- 10-15 млн. опер/сек.,
- 8 конвейерных ФУ,
- 2-х уровневая память.

# История параллельности: матричные процессоры

## ILLIAC IV (1974): матричные процессоры.

Проект: 256 процессорных элементов (ПЭ) = 4 квадранта по 64ПЭ, возможность реконфигурации: 2 квадранта по 128ПЭ или 1 квадрант из 256ПЭ, такт 40нс, производительность 1Гфлоп.

Работы начаты в 1967 году, к концу 1971 изготовлена система из 1 квадранта, в 1974г. она введена в эксплуатацию, доводка велась до 1975 года;

центральная часть: устройство управления (УУ) + матрица из 64 ПЭ.

УУ это простая ЭВМ с небольшой производительностью, управляющая матрицей ПЭ; все ПЭ матрицы работали в синхронном режиме, выполняя в каждый момент времени одну и ту же команду, поступившую от УУ, но над своими данными.

ПЭ имел собственное АЛУ с полным набором команд, ОП - 2Кслова по 64 разряда, цикл памяти 350нс, каждый ПЭ имел непосредственный доступ только к своей ОП.

Сеть пересылки данных: двумерный тор со сдвигом на 1 по границе по горизонтали.

Несмотря на результат в сравнении с проектом: стоимость в 4 раза выше, сделан лишь 1 квадрант, такт 80нс, реальная производительность до 50Мфлоп.

## История параллельности: векторно-конвейерные процессоры

### **CRAY 1 (1976): векторно-конвейерные процессоры**

В 1972 году С.Крэй покидает CDC и основывает свою компанию Cray Research, которая в 1976г. выпускает первый векторно-конвейерный компьютер CRAY-1: время такта 12.5нс, 12 конвейерных функциональных устройств, пиковая производительность 160 миллионов операций в секунду, оперативная память до 1Мслова (слово - 64 разряда), цикл памяти 50нс. Главным новшеством является введение векторных команд, работающих с целыми массивами независимых данных и позволяющих эффективно использовать конвейерные функциональные устройства.

# Основные понятия распараллеливания: уровни параллелизма и гранулярность

## *Уровни параллелизма:*

- **Уровень заданий** – несколько независимых заданий одновременно выполняются на разных процессорах.
- **Уровень программ** – части одной задачи выполняются на множестве процессоров.
- **Уровень команд** – разные фазы нескольких команд выполняются одновременно на различных стадиях конвейера.
- **Уровень данных машинной команды** – бит-последовательные и бит-параллельные операции; параллельно-последовательная обработка; обработка нескольких операндов в одной команде.

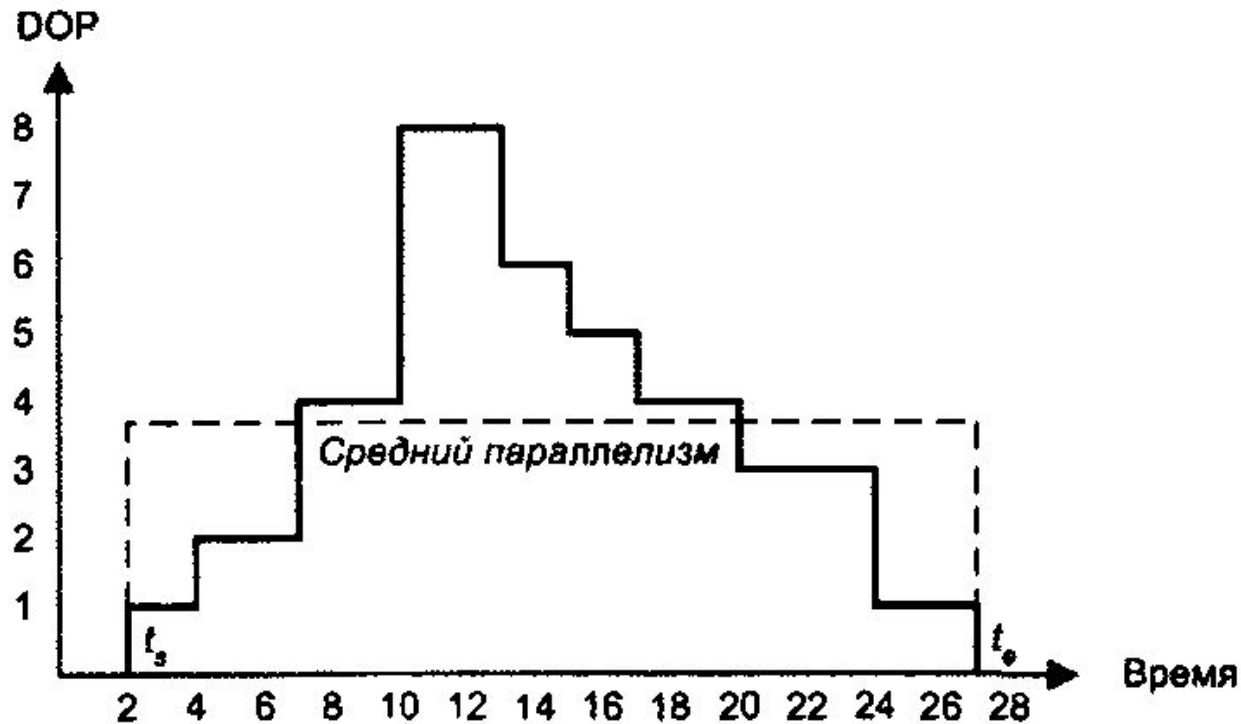
**Гранулярность** – отношение объема вычислений к объему коммуникаций между параллельными ветвями.

- **Крупнозернистый параллелизм** – слабая зависимость между ветвями параллельных вычислений (тысячи исполненных команд на одну операцию обмена).
- **Среднезернистый параллелизм** – средняя зависимость (сотни команд на одну операцию обмена).
- **Мелкозернистый параллелизм** – единицы и десятки команд обработки на одну операцию обмена между параллельными ветвями.

# Основные метрики параллелизма: профиль параллелизма

**Степень параллелизма** (DOP – Degree of Parallelism) – это число участвующих в вычислениях процессоров.

**Профиль параллелизма программ** - изменение во времени степени параллелизма.



# Основные понятия распараллеливания: общий объем вычислительной работы и средний параллелизм

*Общий объем вычислительной работы*  $W$  в интервале времени от стартового момента  $t_s$  до момента завершения  $t_e$  пропорционален площади под кривой профиля:

$$W = T * \int D(t) dt \approx T * \Sigma D_i,$$

где  $T$  – длительность кванта времени, интервал интегрирования и множество индексов  $i$  определяются из  $t_s$  и  $t_e$

*Средний параллелизм*

$$A = ( \int D(t) dt ) / (t_e - t_s) \approx ( \Sigma D_i ) / m,$$

где  $m = (t_e - t_s) / T$

# Основные понятия распараллеливания: Степень ускорения - Speedup

*Ускорение* (speedup) за счет параллельного выполнения:

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

где  $T(1)$  – время исполнения в однопроцессорной системе (число квантов времени),  
 $T(n)$  – время в системе с  $n$  процессорами.

Обычно  $S(n) \leq n$ . При  $S(n) = n$  говорят, что алгоритм показывает *линейное ускорение*.

Причины, по которым  $S(n) > n$ :

а) декомпозированные данные для размещения на процессорах могут иметь меньшее время доступа в силу свойств методов адресации, либо за счет уменьшения количества кэш-промахов;

б) в алгоритме имеются зависимости, обеспечивающие прекращение вычислений до завершения обработки всех данных (например, конъюнкция или дизъюнкция предикатов с существенно различающимися временем вычисления для случаев получения истинного или ложного значения).

Ускорение ограничивают: программные издержки, издержки из-за дисбаланса загрузки процессоров, коммуникационные издержки.



# Основные понятия распараллеливания: эффективность и избыточность

**Эффективность** (efficiency)  $n$ -процессорной системы – ускорение, приходящееся на один процессор:  $E(n) = S(n)/n = T(1)/(n * T(n))$ . Обычно имеет место:  $1/n \leq E(n) \leq 1$ .

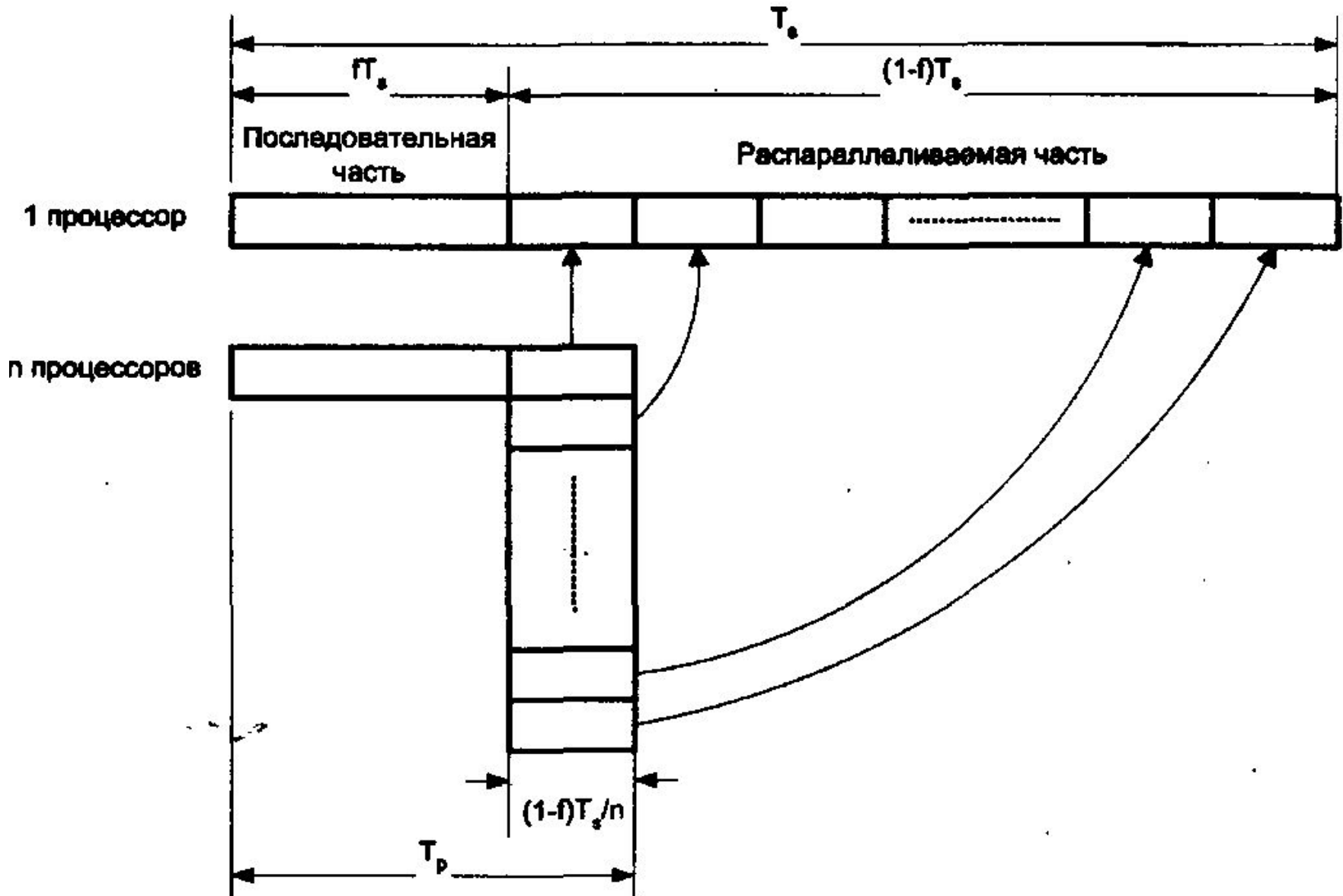
**Избыточность** (redundancy) – величина, учитывающая накладные расходы на организацию параллельных вычислений в среде с  $n$  процессорами:

$$R(n) = O(n)/O(1) = 1/E(n) - 1,$$

где  $O(n)$  и  $O(1)$  – число операций, выполненных соответственно в среде с  $n$  процессорами и в однопроцессорной среде.

Справедливо неравенство  $1 \leq R(n) \leq n$ .

# Распараллеливание: Закон Амдала (1967)



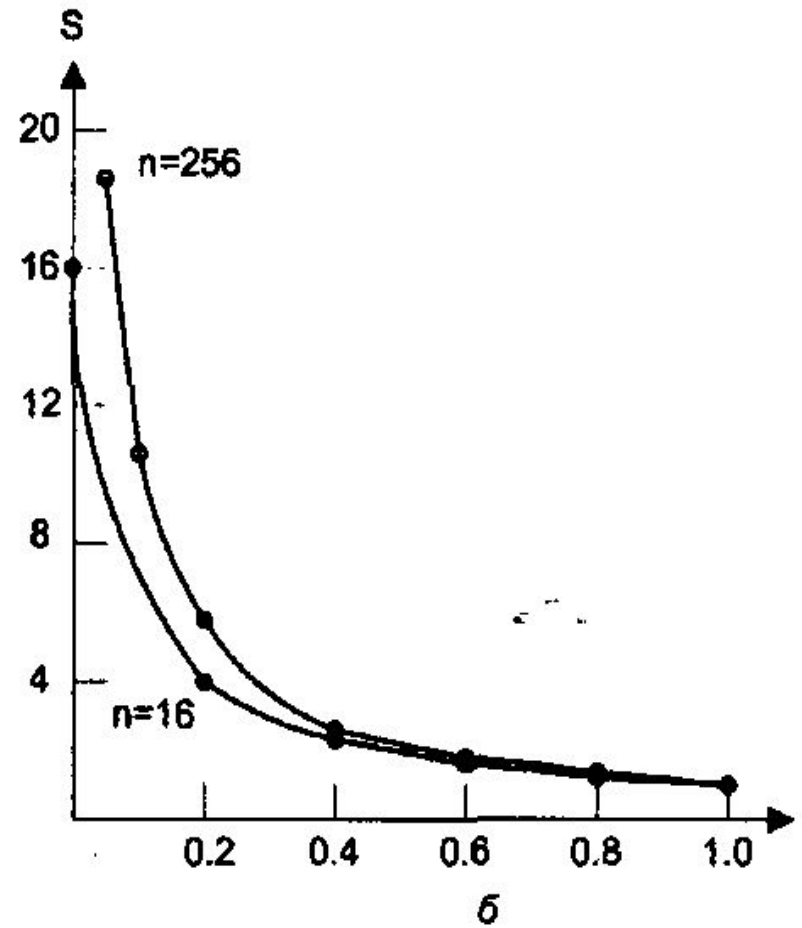
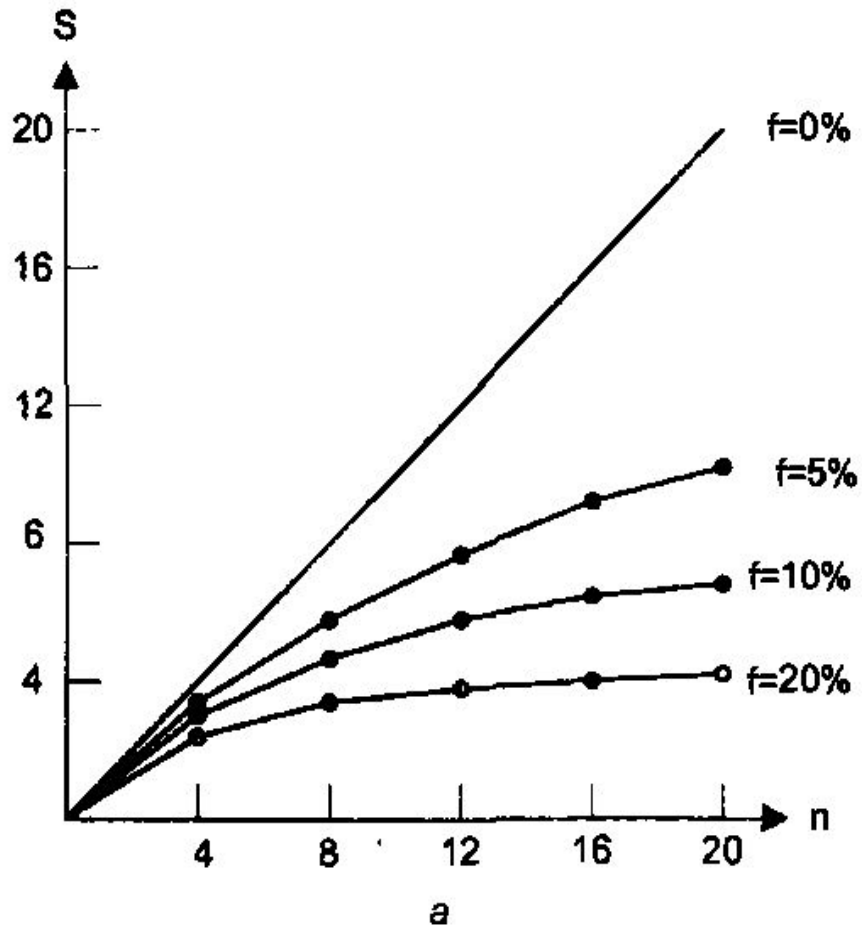
## Распараллеливание: Закон Амдала

$$T_p = f \times T_s + \frac{(1-f) \times T_s}{n}.$$

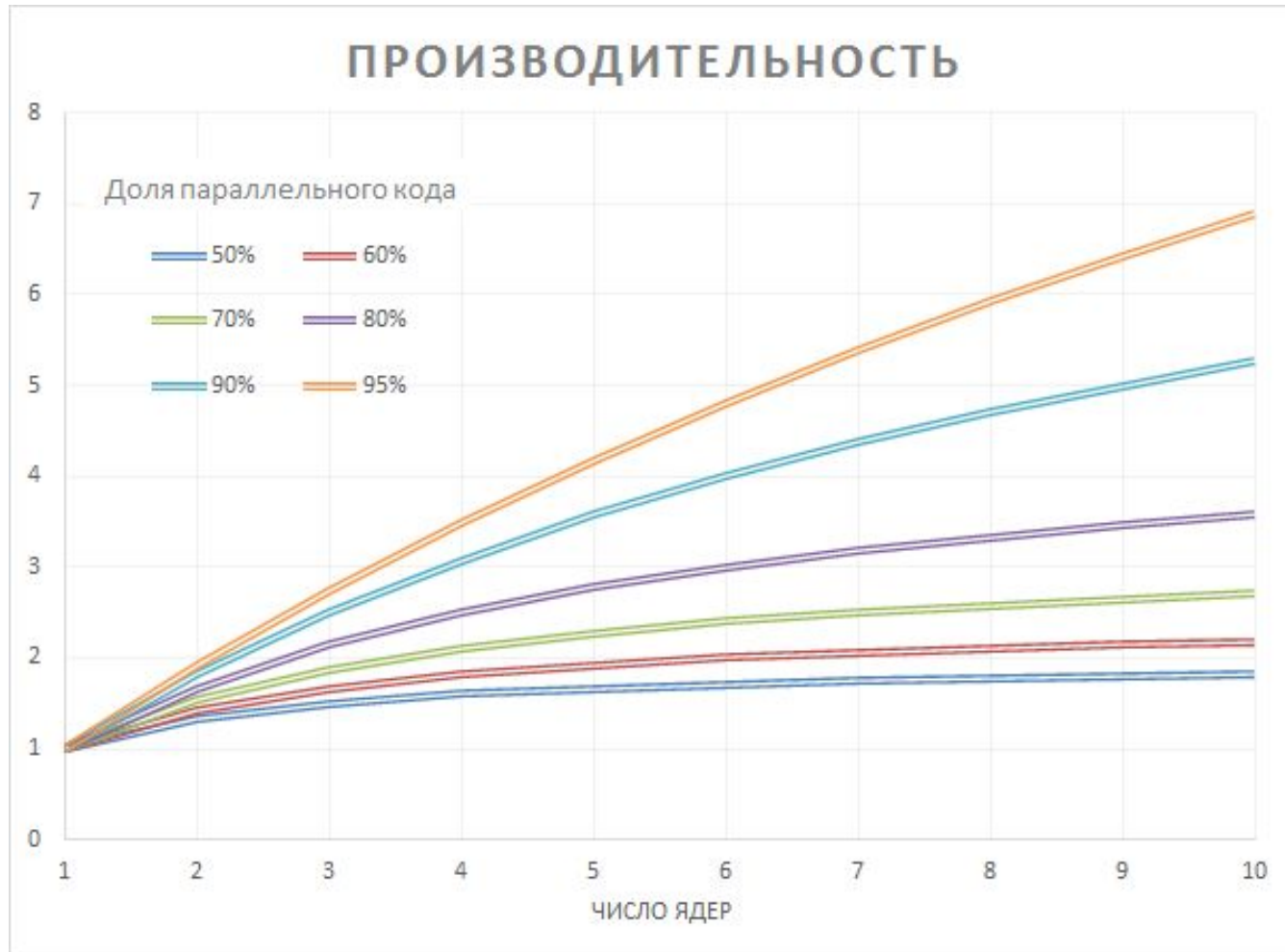
$$S = \frac{T_s}{T_p} = \frac{n}{1 + (n-1) \times f}.$$

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

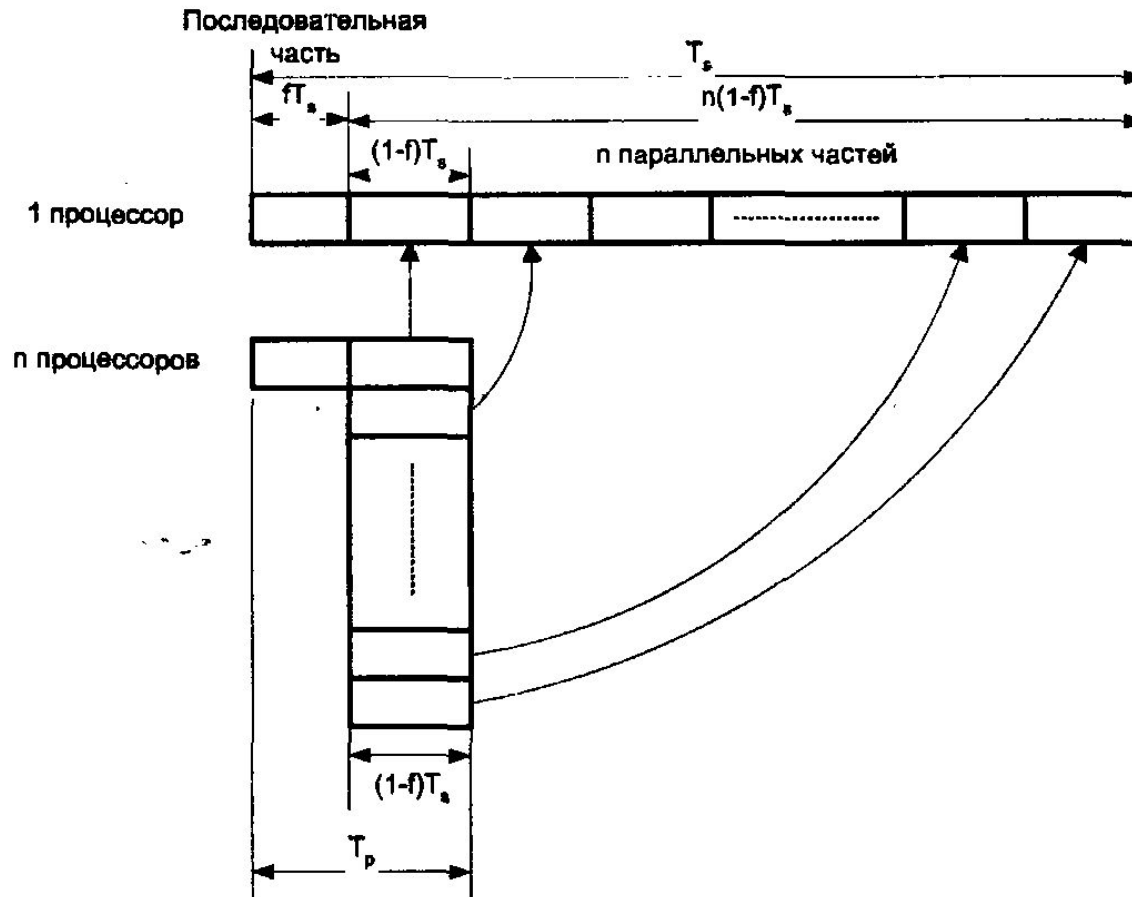
# Распараллеливание: Закон Амдала



# Распараллеливание: Закон Амдала – зависимость ускорения от доли $f$ и $n$



# Распараллеливание: Закон масштабирования ускорения Густафсона-Барсиса



# Распараллеливание: Закон масштабируемого ускорения Густафсона- Барсиса

$$S = \frac{T_s}{T_p} = \frac{f \times T_s + n \times (1 - f) \times T_s}{f \times T_s + (1 - f) \times T_s} = n + (1 - n) \times f.$$

# Распараллеливание: Сопоставление ускорения по Амдалу и по Густафсону-Барсису

Пусть доля последовательной части  $f = 0,1$  // 10%

N	2 ядра	4 ядра	1 класс: 12 2- ядерных & 4- ядерный сервер	5 классов	15 классов
S_Amdal	1.82	3.08	7.57	8.62	9.79
S_Gustafs	1.90	3.70	25.30	50.50	378.10

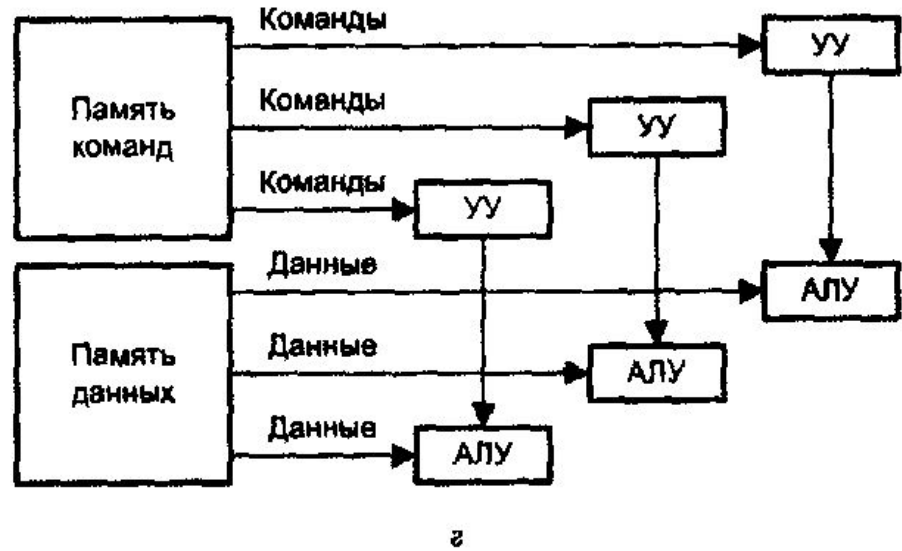
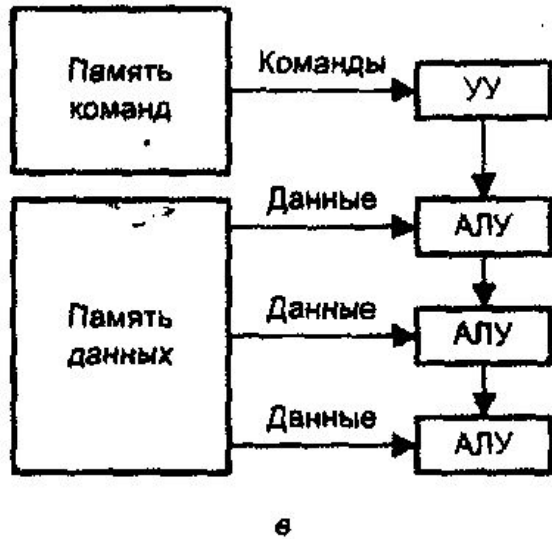
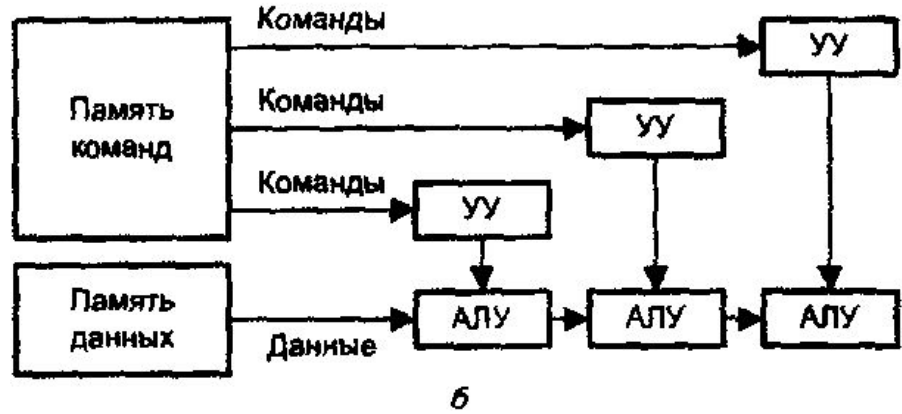
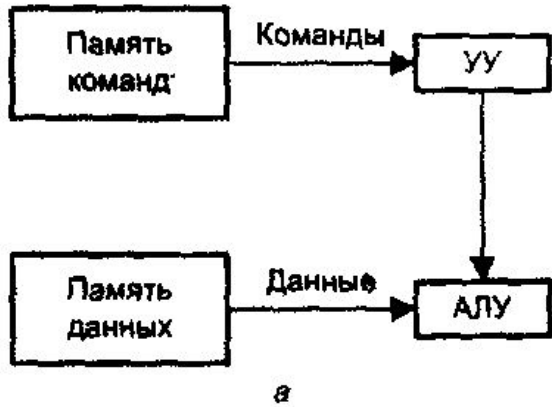


# Классификация параллельных систем по Флинну

Профессор Стенфорда Майкл Флинн в 1966 году предложил классифицировать параллельные системы по наличию параллельности в потоках команд потоках обрабатываемых данных, а в 1972 добавлено еще основание по типу организации памяти:

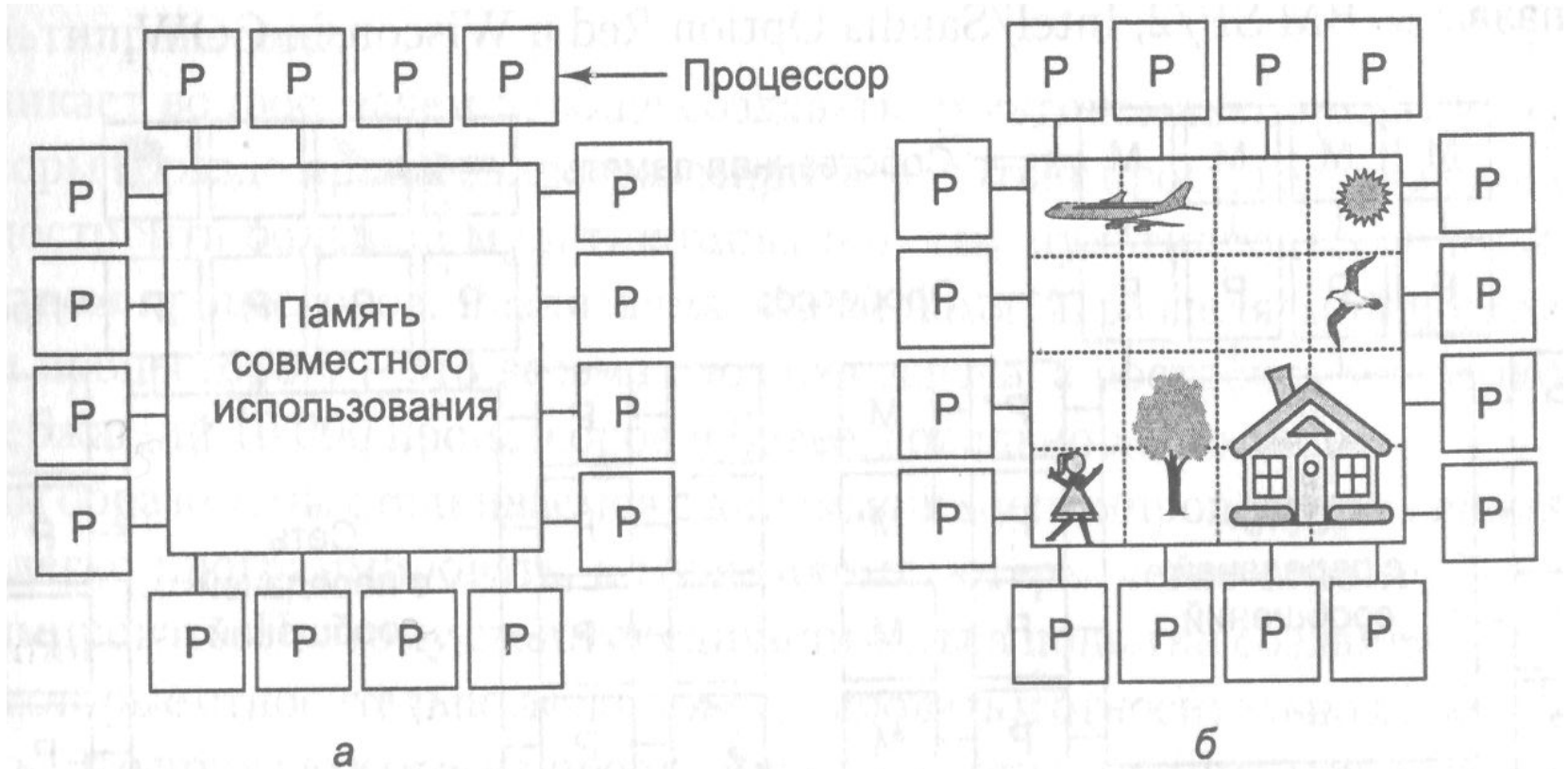
- **SISD: Single Instruction & Single Data** – одиночный поток команд и одиночный поток данных;
- **SIMD: Single Instruction & Multiple Data** – одиночный поток команд и множественный поток данных;
- **MISD: Multiple Instruction & Single Data** – одиночный поток команд и одиночный поток данных;
- **MIMD: Multiple Instruction & Multiple Data** – одиночный поток команд и множественный поток данных;
- **SM-MIMD: Shared Memory MIMD** – мультипроцессорные с общей памятью;
- **DM-MIMD: Distributed Memory MIMD** – мультипроцессорные с распределенной памятью

# Архитектура базовых классов параллельных систем



а) SIMD; б) MISD; в) SIMD; г) MIMD

# Архитектура параллельных систем с общей памятью SM-MIMD

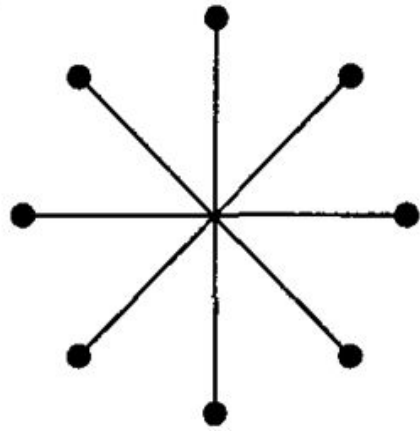


# Архитектура параллельных систем с общей памятью

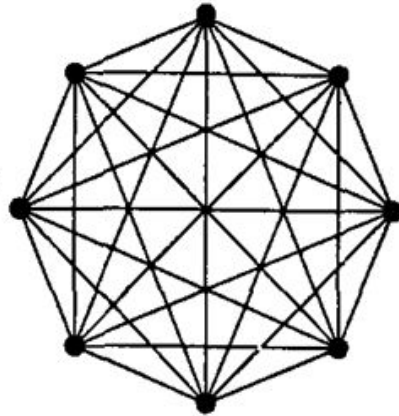
## DM-MIMD



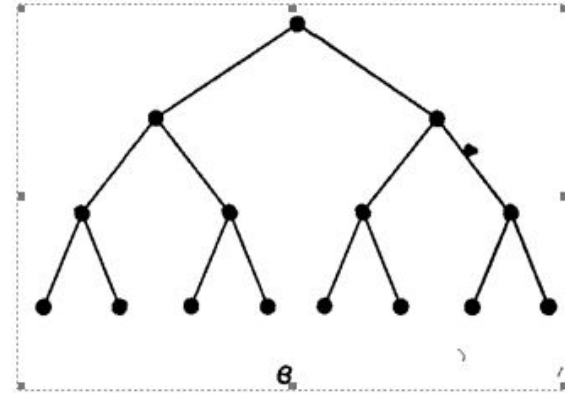
# Организация межсоединений в параллельных системах



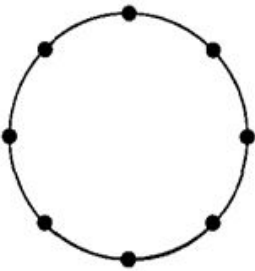
а



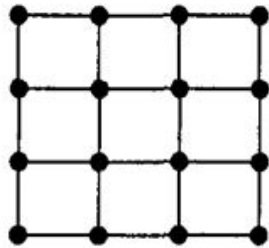
б



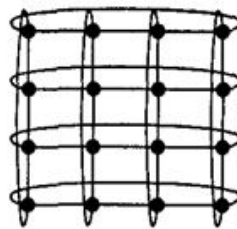
в



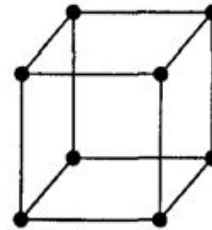
г



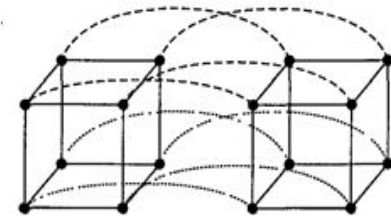
д



е



ж



з

Различные топологии: а) звезда; б) полное межсоединение (full interconnect); в) дерево; г) кольцо; д) решетка; е) двойной тор; ж) куб; з) гиперкуб