



# Многопоточное программирование

Архитектурная поддержка  
многозадачности

# Параллельная обработка

Сложение векторов

$(a_1, a_2, \dots, a_{100}) + (b_1, b_2, \dots, b_{100})$

5 тактов на операцию сложения

# Параллельная обработка

Сложение векторов

$(a_1, a_2, \dots, a_{100}) + (b_1, b_2, \dots, b_{100})$

5 тактов на операцию сложения

- 1 устройство — 500 тактов

# Параллельная обработка

Сложение векторов

$(a_1, a_2, \dots, a_{100}) + (b_1, b_2, \dots, b_{100})$

5 тактов на операцию сложения

- 1 устройство — 500 тактов
- 2 устройства - 250 тактов

# Параллельная обработка

Сложение векторов

$(a_1, a_2, \dots, a_{100}) + (b_1, b_2, \dots, b_{100})$

5 тактов на операцию сложения

- 1 устройство — 500 тактов
- 2 устройства - 250 тактов
- N устройство -  $500/N$  тактов

# Конвейерная обработка

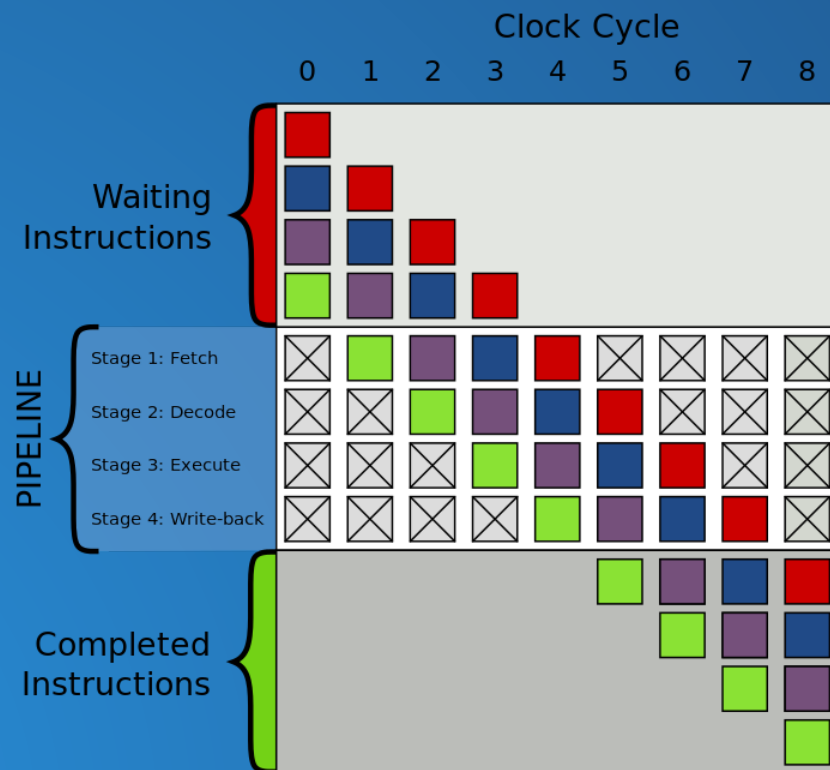
Стадии исполнения инструкции:

- Выборка
- Декодирование
- Исполнение
- Запись результата

Суперскалярные компьютеры

Скалярные и векторные операции

# Конвейерная обработка



# Конвейерная обработка

$n$  — количество элементов в векторе

$\sigma$  — время на инициализацию векторной команды

$l$  — количество ступеней конвейера

$t$  — длительность выполнения ступени конвейера



# Конвейерная обработка

$n$  — количество элементов в векторе

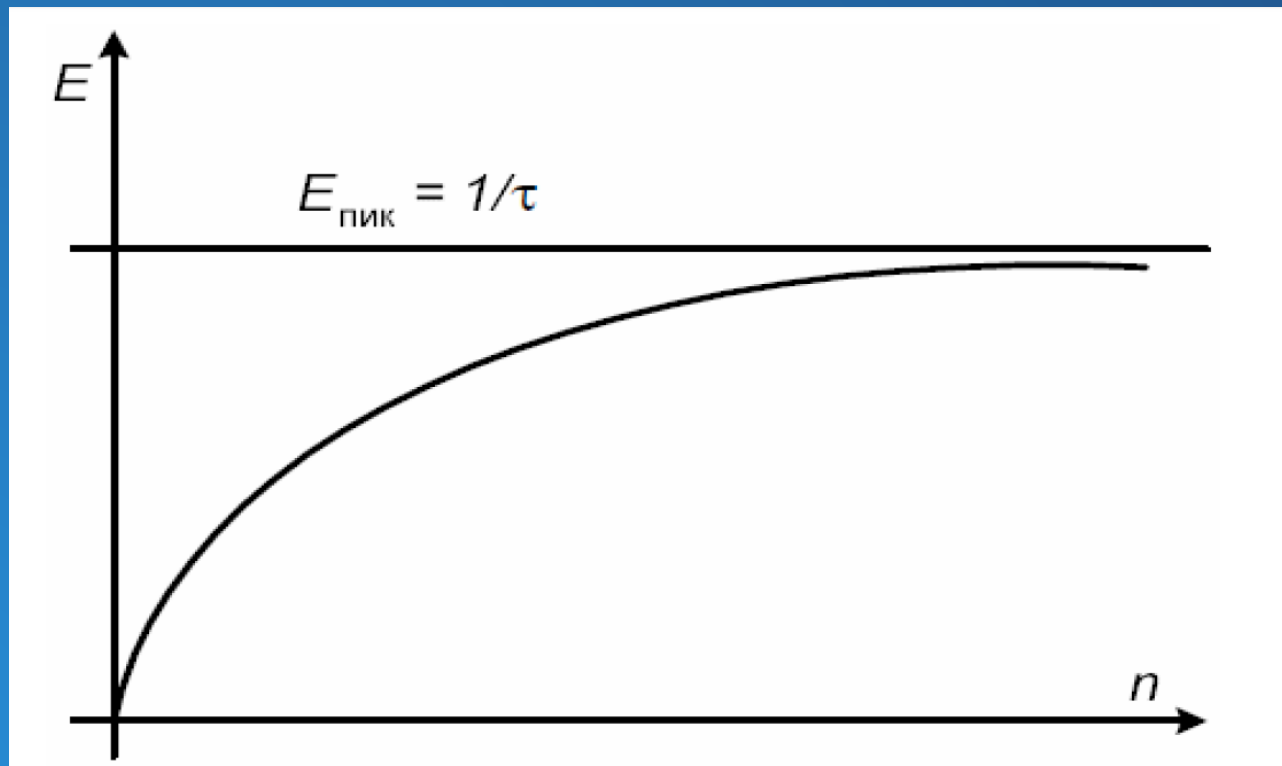
$\sigma$  — время на инициализацию векторной команды

$l$  — количество ступеней конвейера

$\tau$  — длительность выполнения ступени конвейера

$$E = \frac{n}{t} = \frac{n}{[(\sigma + l + n - 1) \tau]} = \frac{1}{\left[ \tau + (\sigma + l - 1) \frac{\tau}{n} \right]}$$

# Конвейерная обработка



# Архитектуры процессоров CISC и RISC

- CISC – Complete Instruction Set Computer
  - окращение количества машинных циклов на выполнение одной инструкции
  - Увеличивалось количество инструкций
  - Использовался микрокод
- RISC – Reduced Instruction Set Computer
  - Сложные инструкции CISC используются редко –инструкции будут простыми
  - Вместо микрокода в ПЗУ (медленно) используется программа в ОЗУ (быстро)
  - Необходимо использование оптимизирующих компиляторов, зато не нужно писать на ассемблере
  - Все инструкции имеют одинаковую длину и состоят из битовых полей, которые можно декодировать одновременно
  - Используется большое количество регистров для сокращения количества обращений к памяти

# Классификация Флинна (1966)

SISD (Single Instruction stream over a Single Data stream)

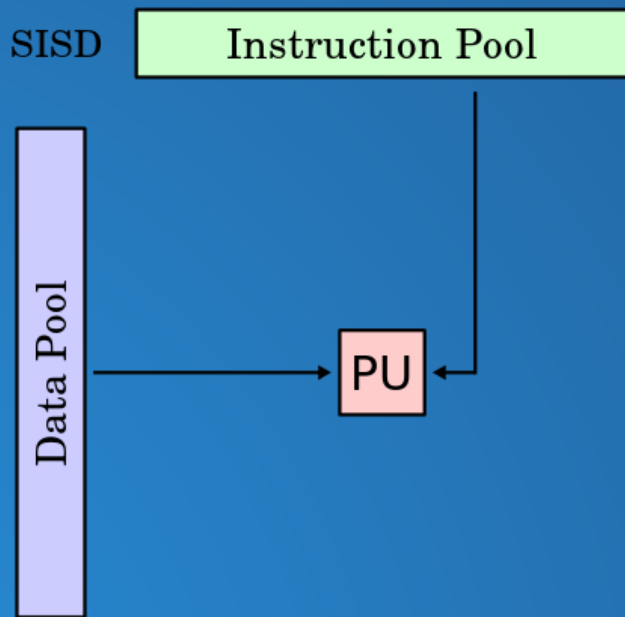
— один поток команд и один поток данных

SIMD (Single Instruction, Multiple Data) — один поток команд и несколько потоков данных

MISD (Multiple Instruction Single Data) — несколько потоков команд и один поток данных

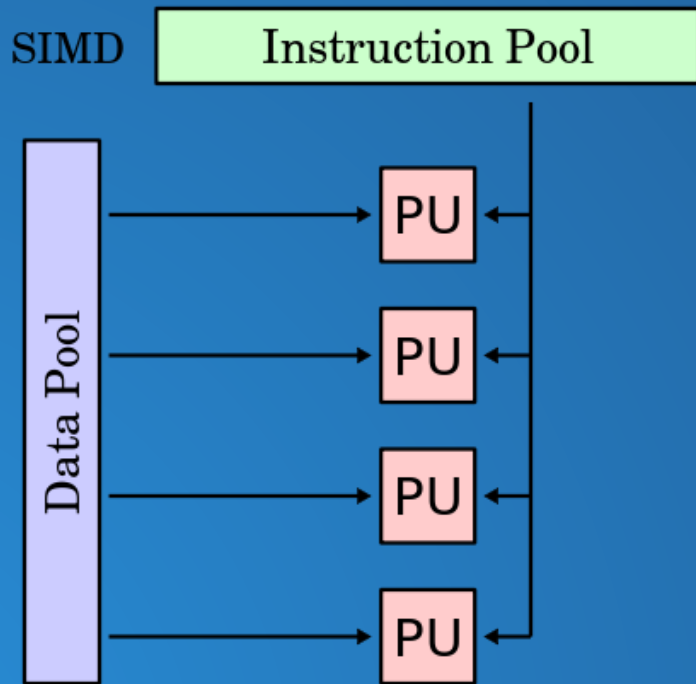
MIMD (Multiple Instruction Multiple Data) — несколько потоков команд и несколько потоков данных

# SISD архитектура



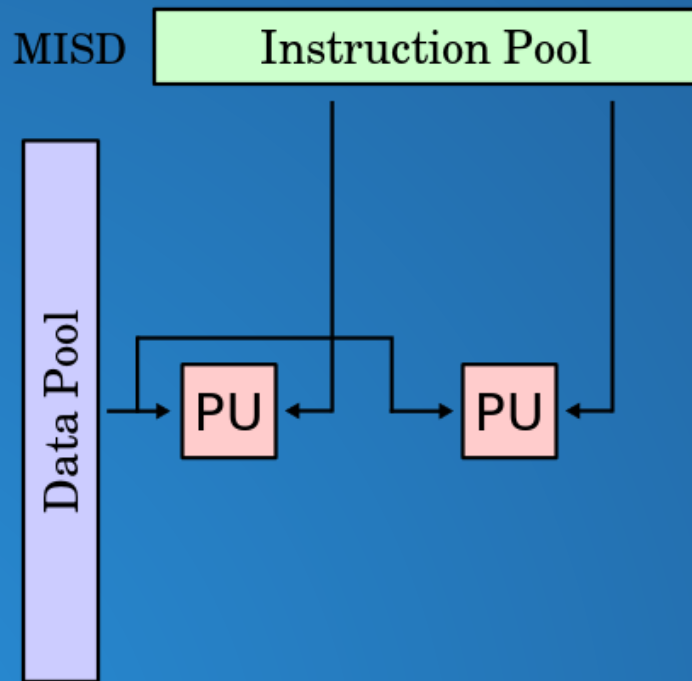
- Традиционные последовательные компьютеры
- Классическая архитектура фон Неймана

# SIMD архитектура

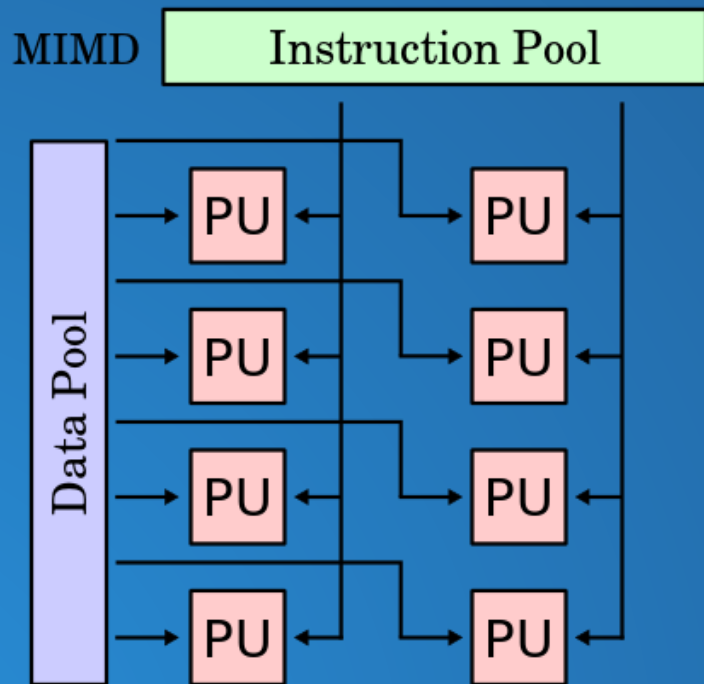


- Векторные процессоры
- Intel MMX, Pentium, Core 2 Duo
- GPU

# MISD архитектура

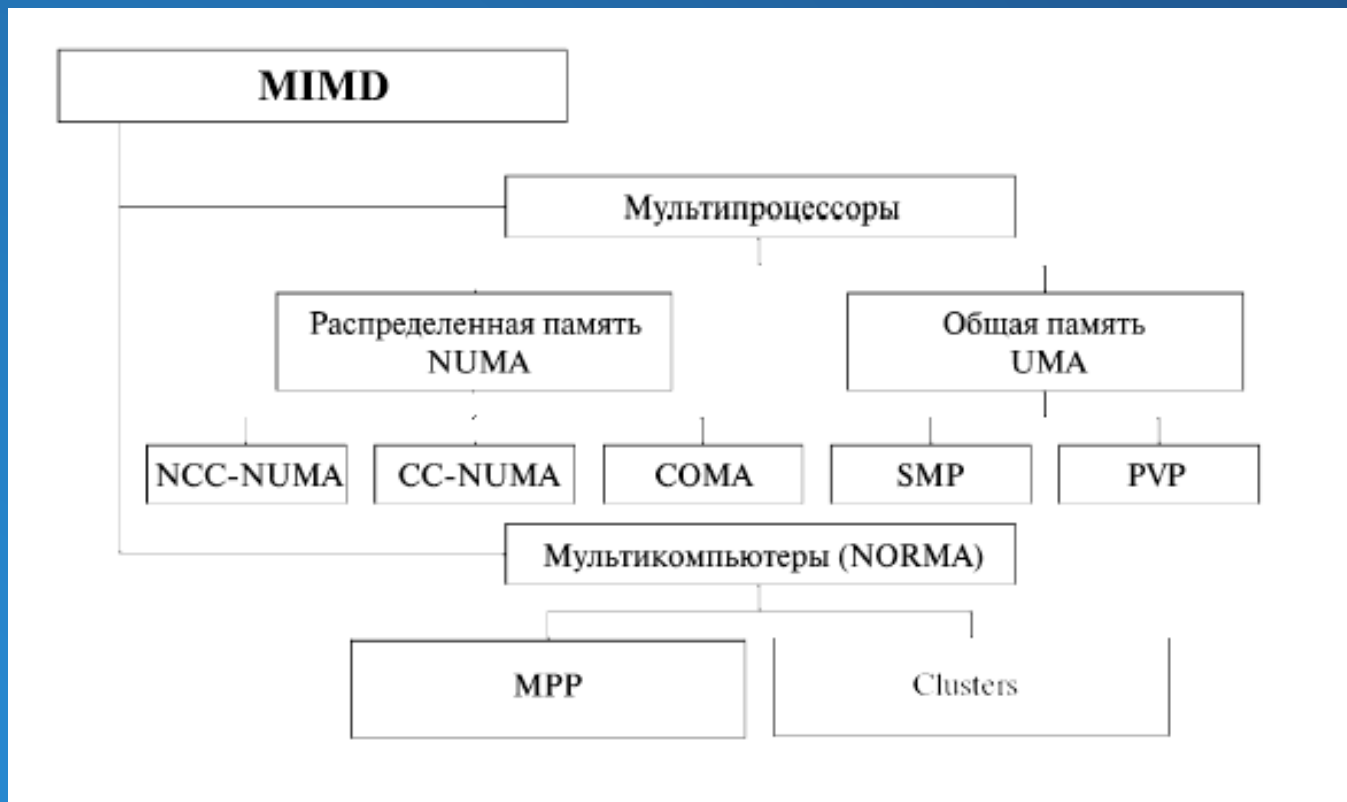


# MIMD архитектура

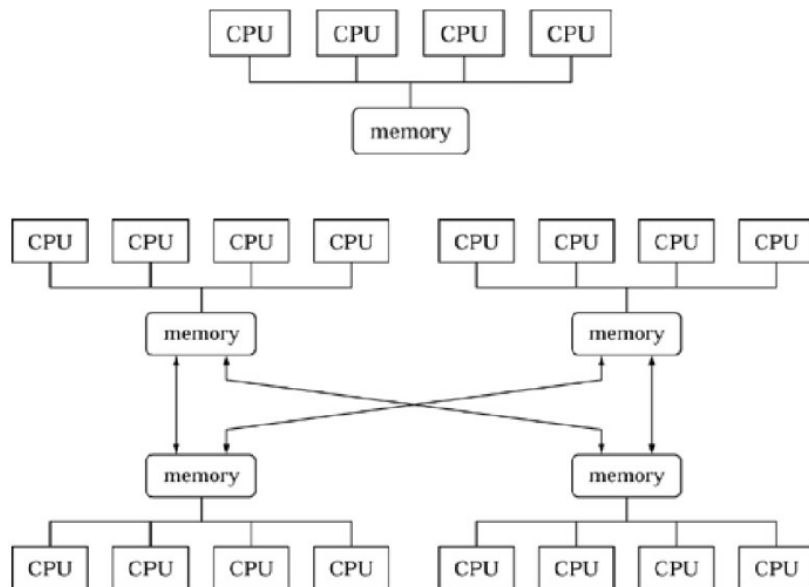




# MIMD архитектура



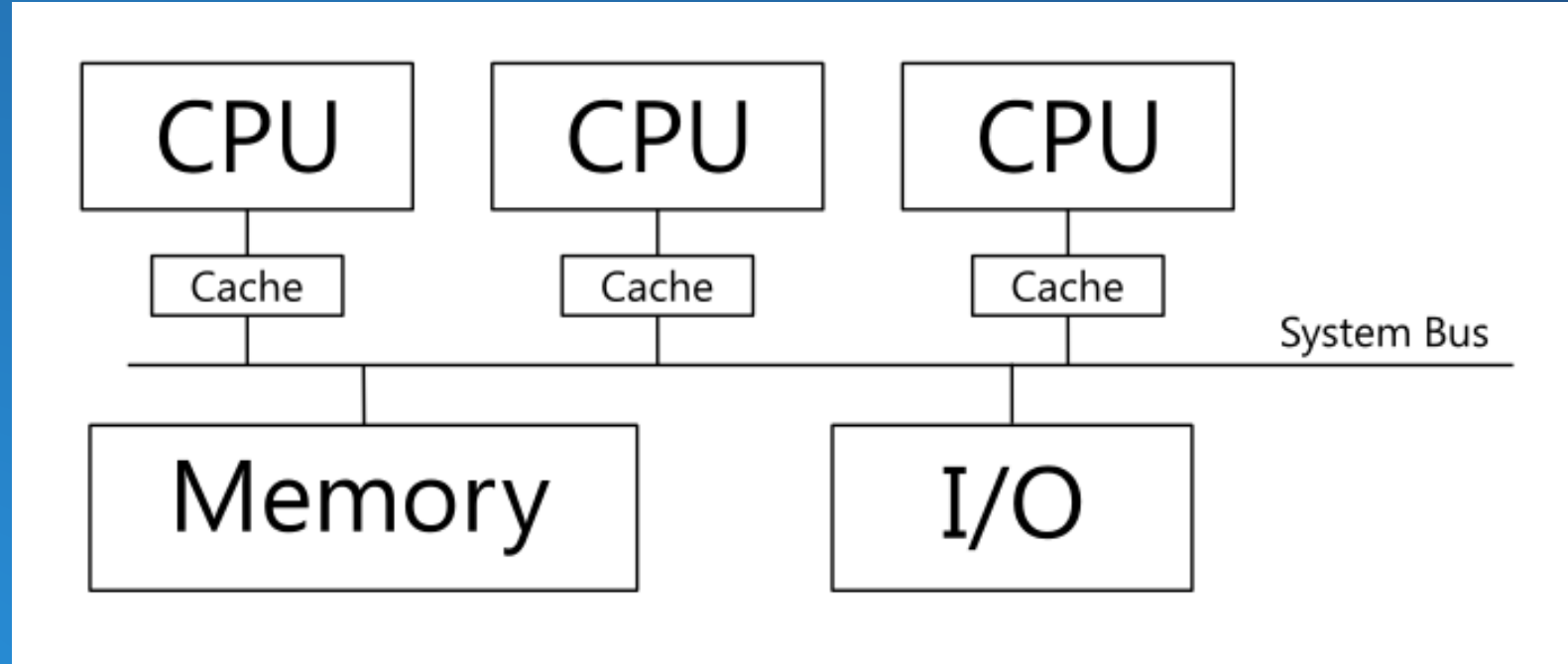
# Мультипроцессоры



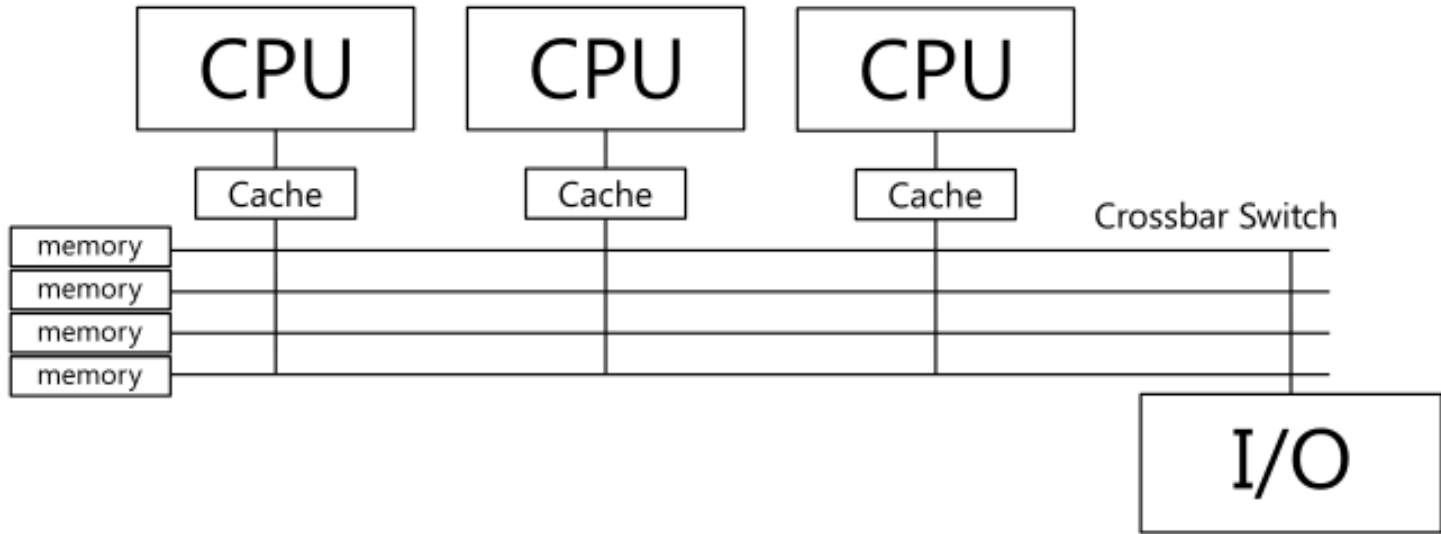
# SMP (Symmetrical Multiprocessing)

- Два или более процессоров подсоединены к общей памяти
- Задачи легко перемещаются между процессорами
- Самый дешевый способ

# SMP (Symmetrical Multiprocessing)



# SMP (Symmetrical Multiprocessing)



# SMP (Symmetrical Multiprocessing)

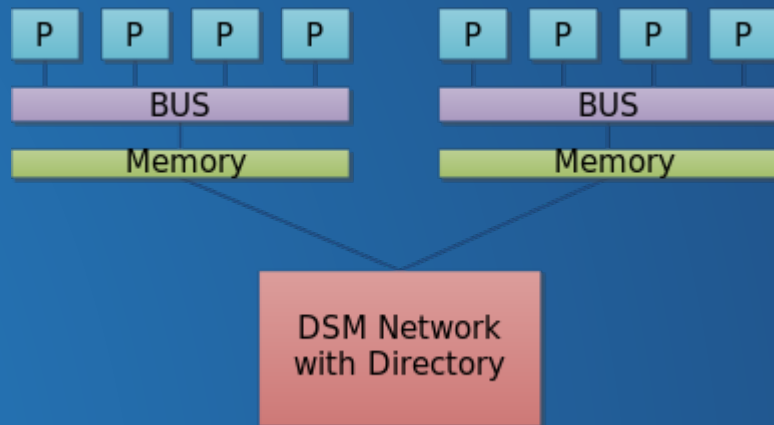
- Архитектура большинства многопроцессорных систем
- Любой процессор над любыми данными
- Легко масштабируется
- Узкое место - общая шина
- Проблема когерентности кэш-памяти

# VLIW - архитектура

- Very Long Instruction Word
- Несколько вычислительных устройств
- Одна инструкция - несколько операций
- Простая архитектура
- Распределение вычислений - задача компилятора

# NUMA (Non-Uniform Memory Access)

- Память физически распределена
- Память логически объединена
- Время доступа к памяти определяется ее расположением





# Мультипроцессоры

## Общая распределяемая память (UMA)

- Преимущества:
  - Привычная модель программирования
  - Высокая скорость обмена данными

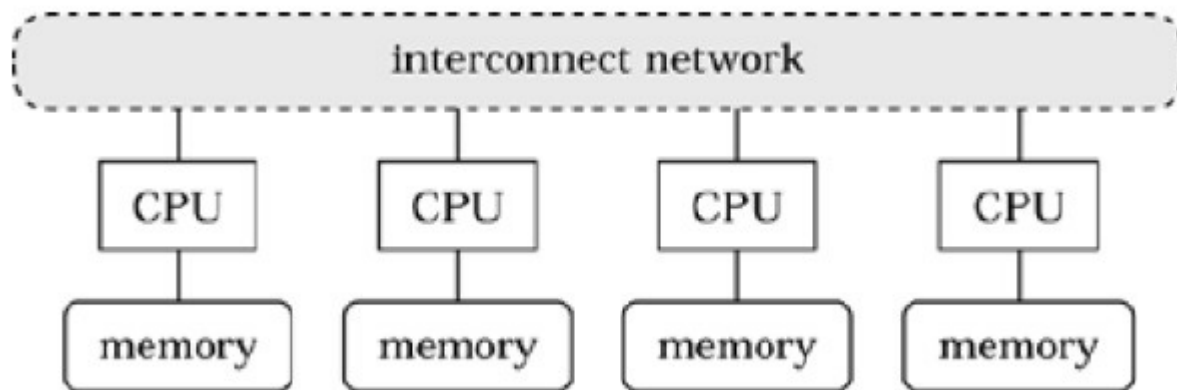
# Мультипроцессоры

## Общая распределяемая память

- Преимущества:
  - Привычная модель программирования
  - Высокая скорость обмена данными
- Недостатки
  - Синхронизация при доступе к общим данным
  - Когерентность кэшей
  - Масштабируемость
  - Эффективное использование память в NUMA

# Мультикомпьютеры

## Распределяемая память



# Мультикомпьютеры

- Массивно-параллельные системы (MPP)
- Кластеры
- Network of workstation(NOW)
- GRID

# Мультикомьютеры

## Распределенная память

- Преимущества
  - Низкая стоимость
  - Высокая масштабируемость
  - Меньше проблем с синхронизацией
  - Декомпозиция на крупные задачи

# Мультикомьютеры

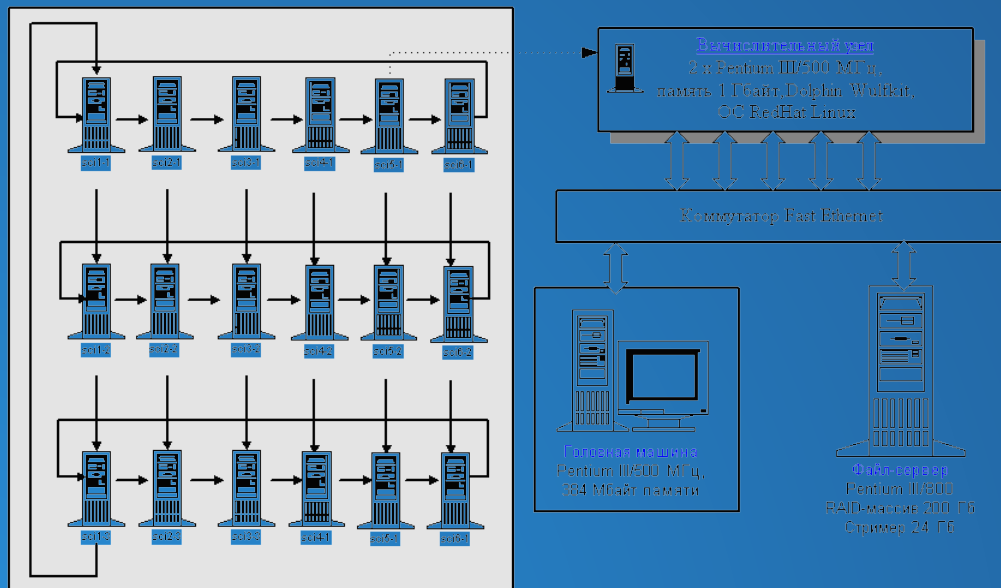
## Распределенная память

- Преимущества
  - Низкая стоимость
  - Высокая масштабируемость
  - Меньше проблем с синхронизацией
  - Декомпозиция на крупные задачи
- Недостатки
  - Необходимость использования сообщений
  - Высокие временные задержки и низкая пропускная способность
  - Отказы узлов

# Кластеры

## Структура вычислительного кластера

36 процессоров Pentium III 500 МГц, Dolphin PCI (двухмерный зрр), произвольные 13 GBLOES



- Несколько связанных между собой компьютеров
- используется как один информационный ресурс

# GRID

- Компьютерная среда, обеспечивающая скоординированное разделение вычислительных ресурсов и ресурсов хранения информации в одной виртуальной организации
- Форма распределенных вычислений, использующая «виртуальный кластер» состоящий из множества неоднородных вычислительных ресурсов
- «метакомпьютинг»



# GRID



# GRID

- Система получает задачи и делегирует вычисления ресурсам зарегистрированным в ней
- Ресурсы могут быть неоднородны и географически распределенными, это могут быть как производительные кластеры, так и обычные рабочие станции

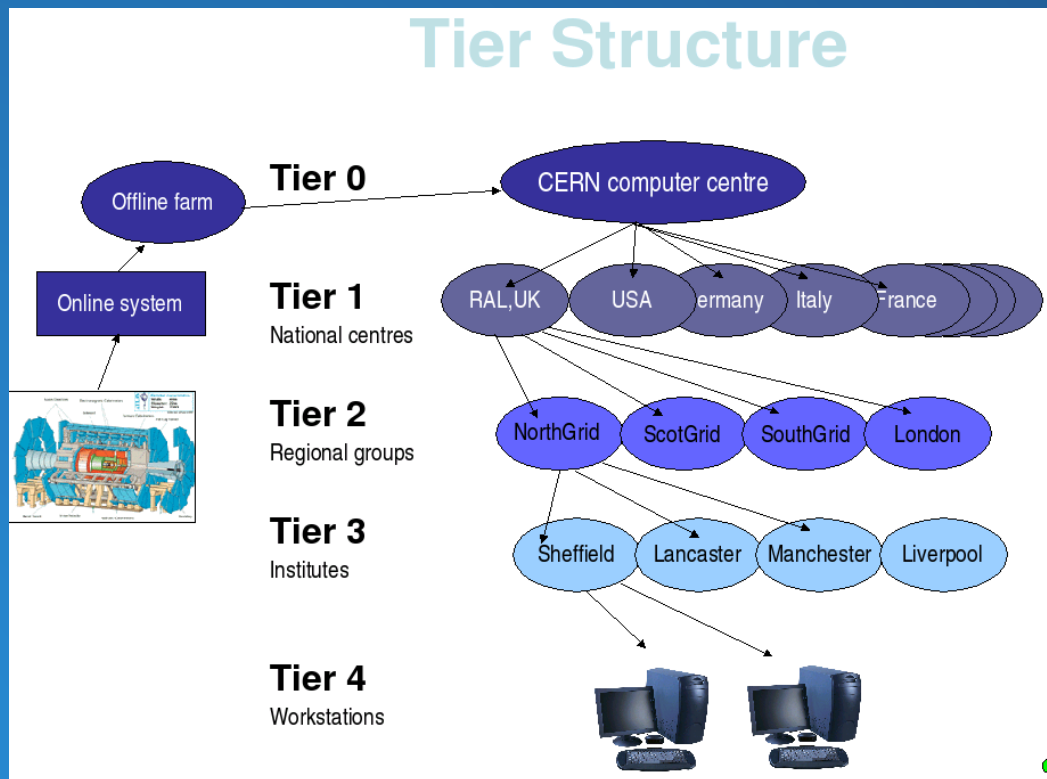
# CERN GRID

GRID имеет иерархическую структуру

Пример: GRID система CERN

- Верхний уровень Tier 0 — получение информации с детекторов, сбор «сырых» научных данных, которые будут храниться до конца работы эксперимента
- Tier 1 — хранение второй копии этих данных в других уголках мира, 11 центров по всему миру, Центры обладают значительными ресурсами для хранения данных
- Tier 2 — следующие в иерархии, многочисленные центры второго уровня, региональные центры
- Tier 3 — Институты
- Tier 4 — Рабочие станции

# CERN GRID



# GRID

- Менеджер заданий
- Брокер ресурсов
- Система составления расписания
- Хранилище входных данных и результатов

# GRID

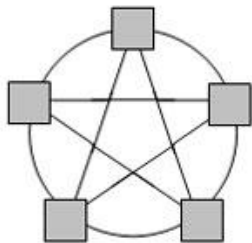
## Применение:

- Научные проекты
  - Обработка результатов работы БАК
  - Поиск лекарств от болезни Альцгеймера и рака
  - Поиск инопланетной жизни
- Коммерческие проекты
  - Прогнозирование курсов валют
  - Расчёт конструкций в авиации и строительстве

# Типовые схемы коммуникации

- Полный граф
- Линейка
- Кольцо
- Звезда
- Решетка
- Гиперкуб

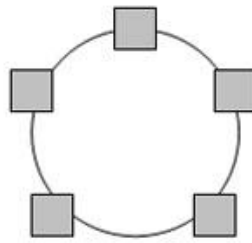
# Типовые схемы коммуникации



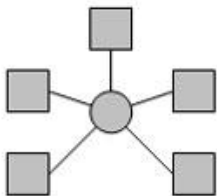
1) Полный граф



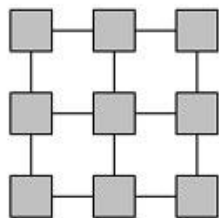
2) Линейка



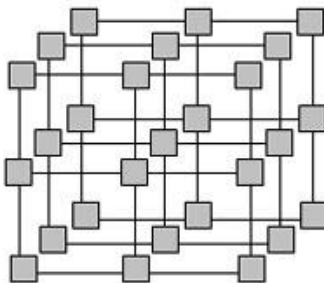
3) Кольцо



4) Звезда



5) 2-мерная решетка



6) 3-мерная решетка



# Топология сети кластеров

- Switch
- Полный граф
- В каждый момент один процессор может участвовать только в одной приемо-передаче сообщения

# Характеристики топологии сети

- Диаметр(максимальное расстояние)
- Связность (разные маршруты)
- Ширина бинарного деления (минимальное число дуг для несвязности )
- Стоимость (количество линий передачи)

# Характеристики топологии сети

Топология	Диаметр	Связность	Ширина бисекции	Стоимость
Полный граф				
Звезда				
Двоичное дерево				
Линейка				
Кольцо				
Решетка $N=2$				
Решетка тор $N=2$				
Гиперкуб				

# Характеристики топологии сети

Топология	Диаметр	Связность	Ширина бисекции	Стоимость
Полный граф	1	$p^2/4$	$p-1$	$p(p-1)/2$
Звезда	2	1	1	$p-1$
Двоичное дерево	$2\log((p+1)/2)$	1	1	$p-1$
Линейка	$p-1$	1	1	$p-1$
Кольцо	$p/2$	2	2	$p$
Решетка $N=2$	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
Решетка тор $N=2$	$2\sqrt{p/2}$	$2\sqrt{p}$	4	$2p$
Гиперкуб	$\log p$	$p/2$	$\log p$	$(p \log p)/2$

# Гибридные системы



# Классификация Хокни



# Классификация Фенга (1972)

- Две простые характеристики:
  - $n$  - количество бит в машинном слове, обрабатываемых параллельно
  - $m$  - число слов обрабатываемых одновременно
- Максимальная степень параллелизма
  - $P = m \times n$
- Преимущества:
- Недостатки:

# Классификация Фенга (1972)

- Две простые характеристики:
  - $n$  - количество бит в машинном слове, обрабатываемых параллельно
  - $m$  - число слов обрабатываемых одновременно
- Максимальная степен параллелизма
  - $P = m \times n$
- Преимущества:
  - едина числовая характеристика
- Недостатки:
  - Нет различия между разными системами
  - Не учитывается например возможное применение конвейера



# Классы по Фенгу

- Разрядно-последовательные, пословно-последовательные ( $n = m = 1$ )
- Разрядно-параллельные, пословно-последовательные ( $n > 1, m = 1$ )
- Разрядно-последовательные, пословно-параллельные ( $n = 1, m > 1$ )
- Разрядно-параллельные, пословно-параллельные ( $n > 1, m > 1$ )

# Другие классификации

Классификация Хендлера

Классификация Шнайдера

Классификация Скилликорна



# Производительность параллельных компьютеров

- Пиковая производительность
- MIPS - Million Instruction per Second
- FLOP - Float Point Operation per Second

# Производительность параллельных компьютеров

## Benchmark

LINPACK - решение систем уравнений

STREAM - работа с очень длинными векторами

# Вопросы

