

Центр компьютерной физики

Кафедра общей физики и волновых процессов

Международный лазерный центр

**Параллельное
программирование
для ресурсоёмких задач численного
моделирования в физике**

*В.О. Милицин, Д.Н. Янышев, И.А.
Буткарев*

Лекция № 1

Содержание курса

- Введение
- Архитектура высокопроизводительных систем
- Особенности параллельных вычислений
- Работа с кластером; операционная система Linux
- Технология OpenMP
- Интерфейс передачи сообщений MPI
- Моделирование физических задач средствами OpenMP и MPI

Структура курса

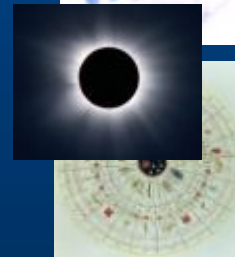
- лекции
- практические занятия
- самостоятельные задания
- ЗАЧЕТ

Развитие инструментов для научного процесса

1. Наблюдение



2. Гипотеза



4. Утверждение

3. Прогноз

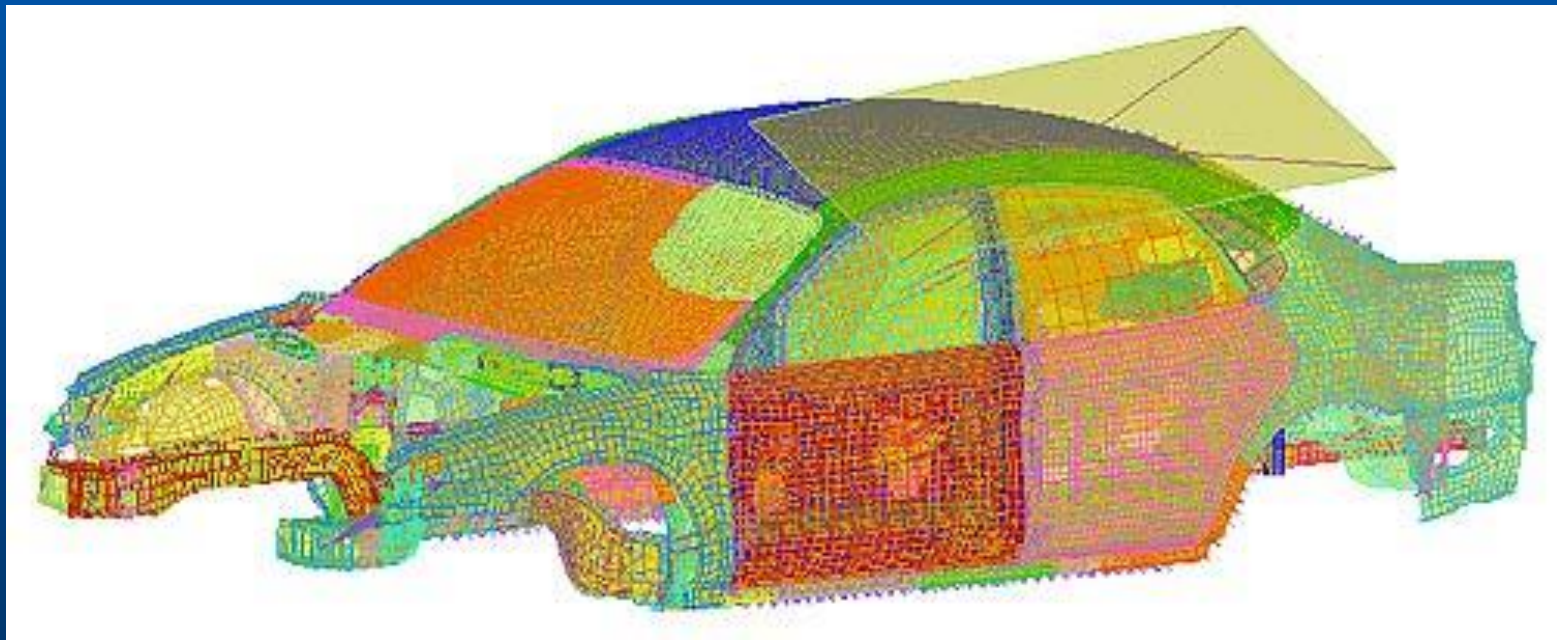
Области применения суперкомпьютеров

- Научно-исследовательские институты
- Университеты
- Геофизика
- Биология
- Метеорология
- Финансовая область
- Энергетика
- Наукоемкие отрасли промышленности
- Медицина
- Генетика
- Геологоразведка
- Контроль за окружающей средой

Пример проблемы решаемой с помощью параллельного программирования

- Car Crash Simulation:
 - Поверхность машины моделируется как считанное количество треугольных элементов.
 - Моделируется позиция каждого элемента во времени с учетом сил действующих на него и его свойств.
 - Каждый элемент взаимодействует только с соседними элементами

Сетка конечных элементов



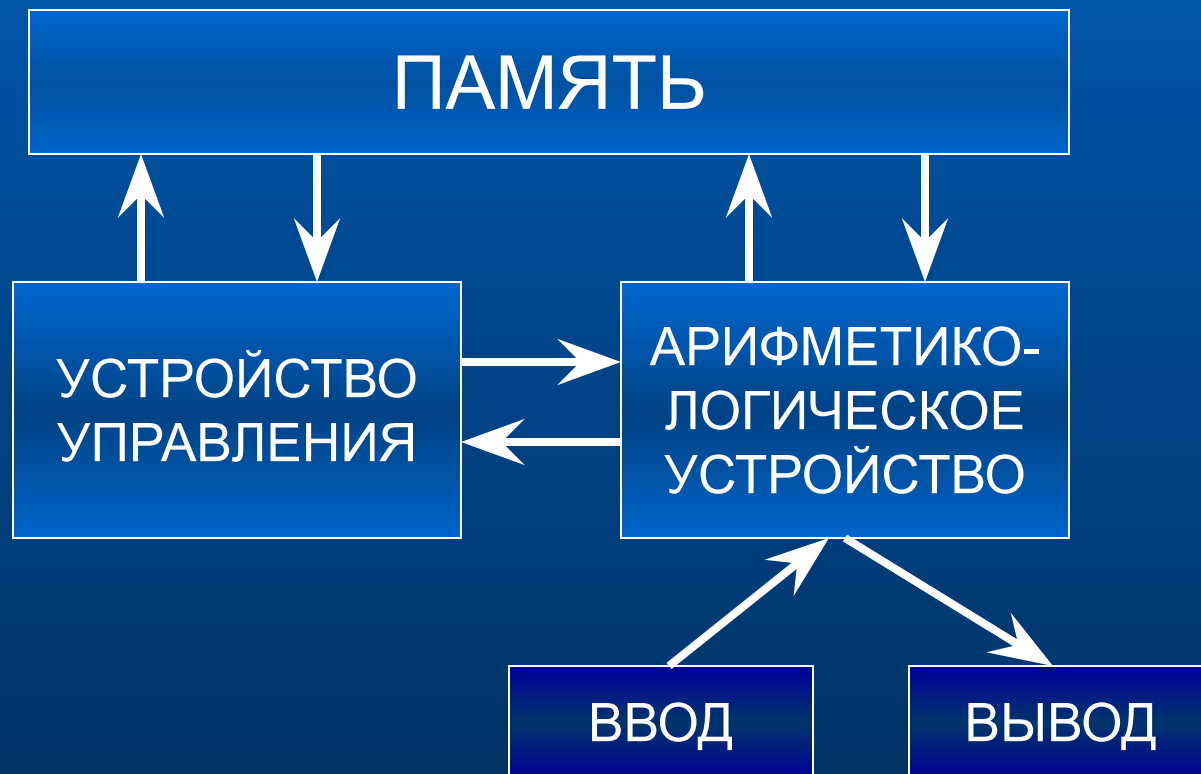
Архитектура ЭВМ

- Архитектура компьютера
- Организация компьютера
- Схема компьютера

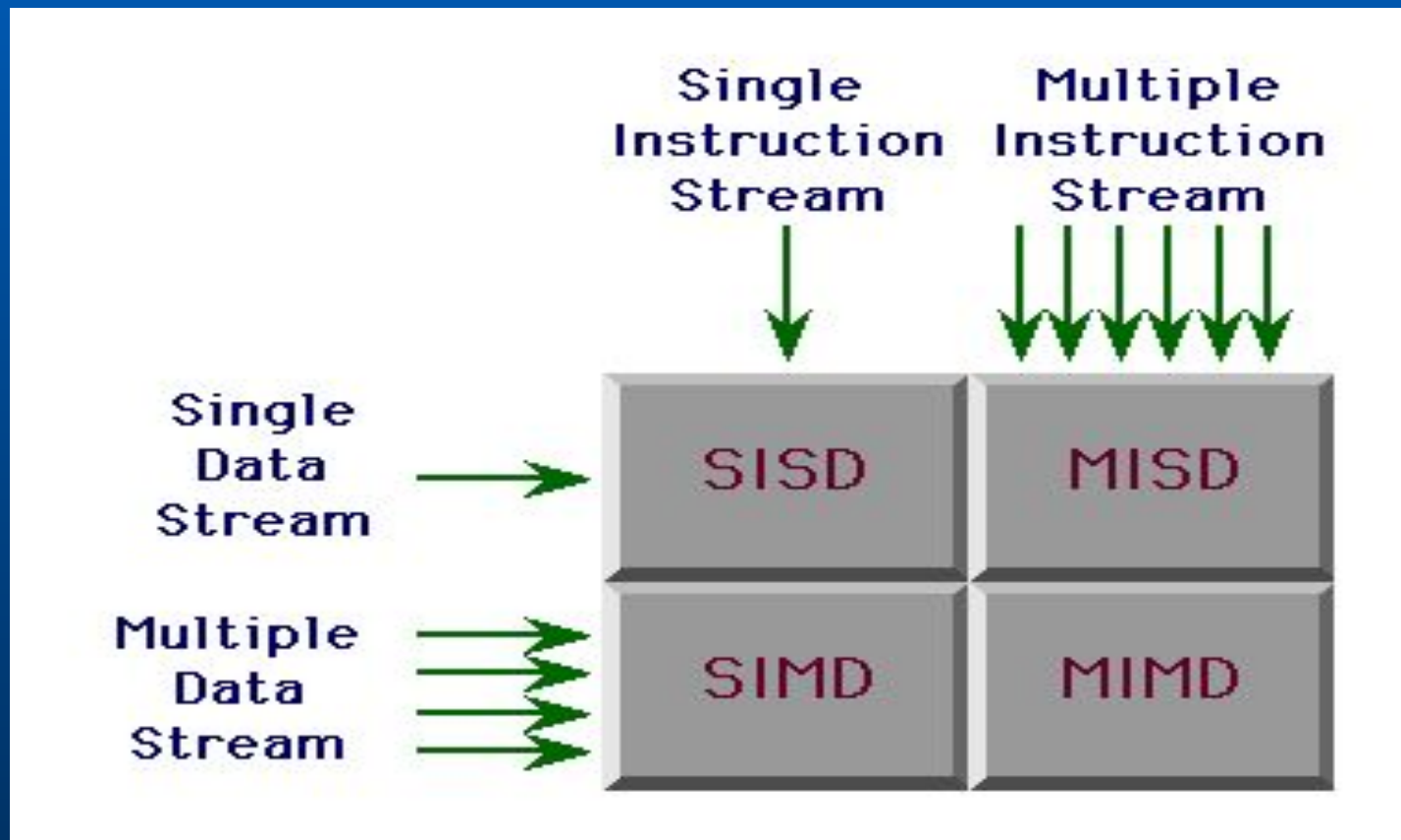
Архитектура фон Неймана

- Программа хранится в компьютере
- Программа во время выполнения и необходимые для её работы данные находятся в оперативной памяти
- Имеется арифметико-логическое устройство, выполняющее арифметические и логические операции с данными
- Имеется устройство управления, которое интерпретирует команды, выбираемые из памяти, и выполняет их
- Устройства ввода и вывода используются для ввода программ и данных и для вывода результатов расчетов. Работают под управлением УУ

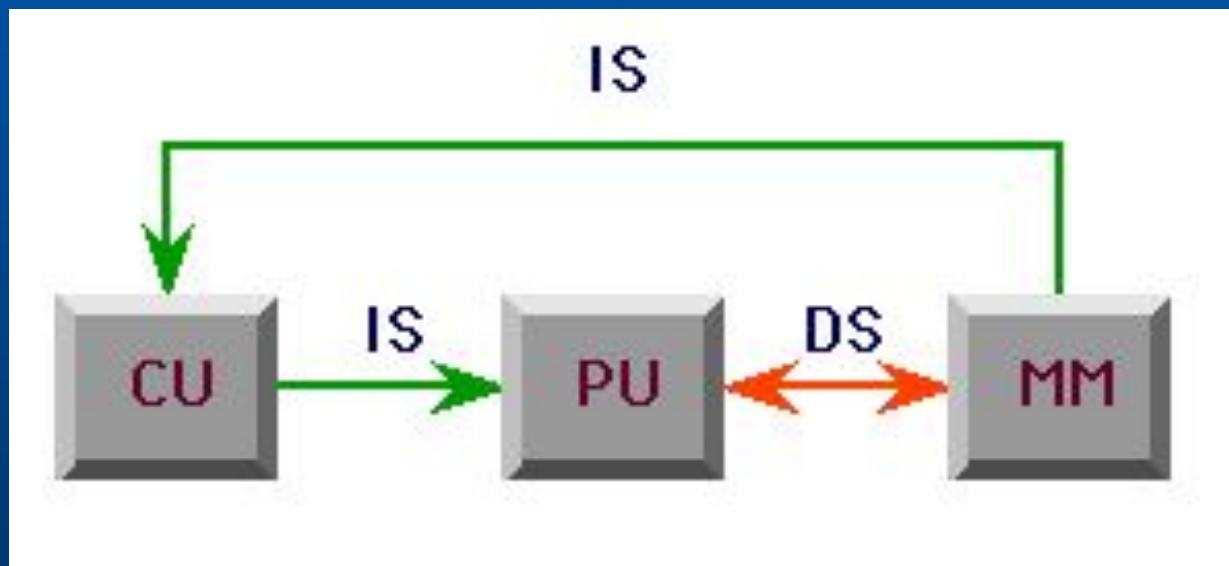
Архитектура фон Неймана II



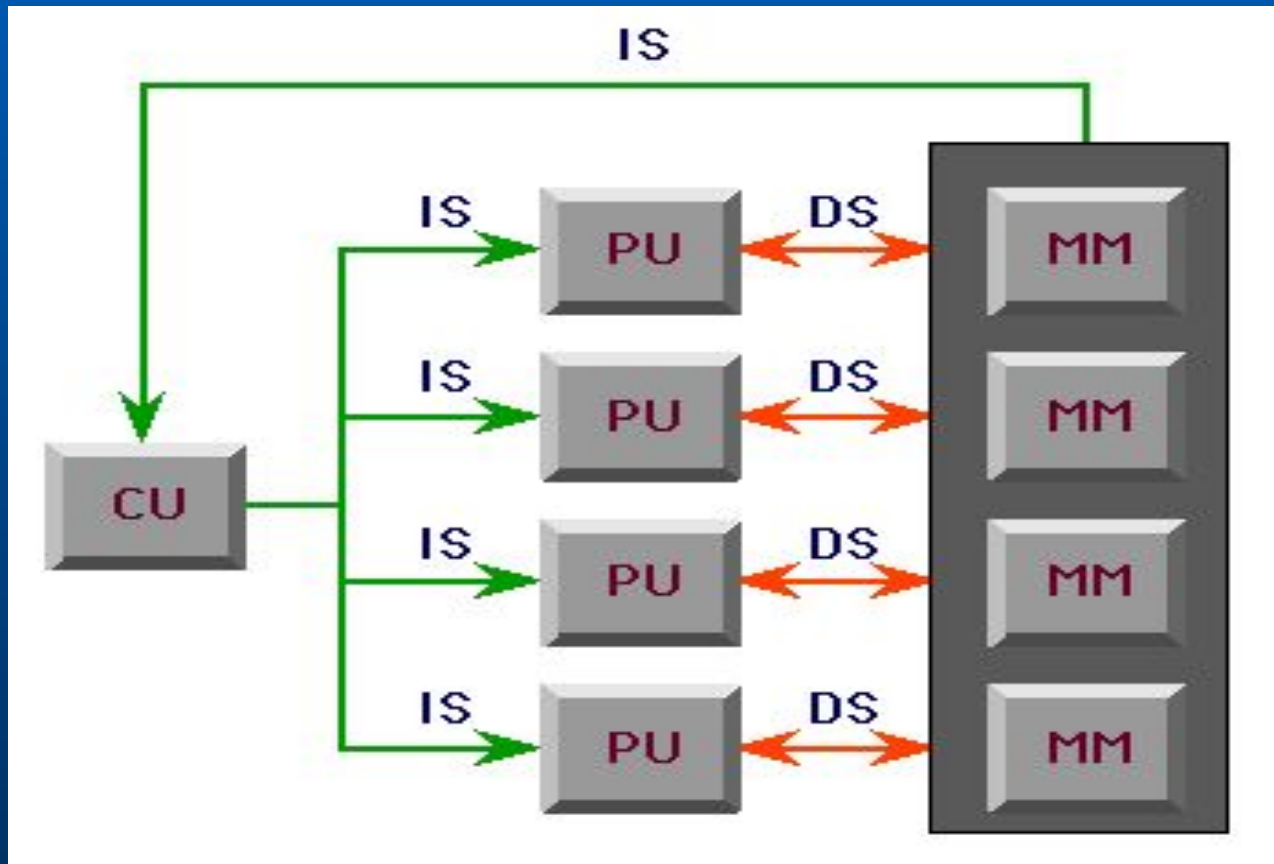
Классификация Флинта



SISD-компьютеры



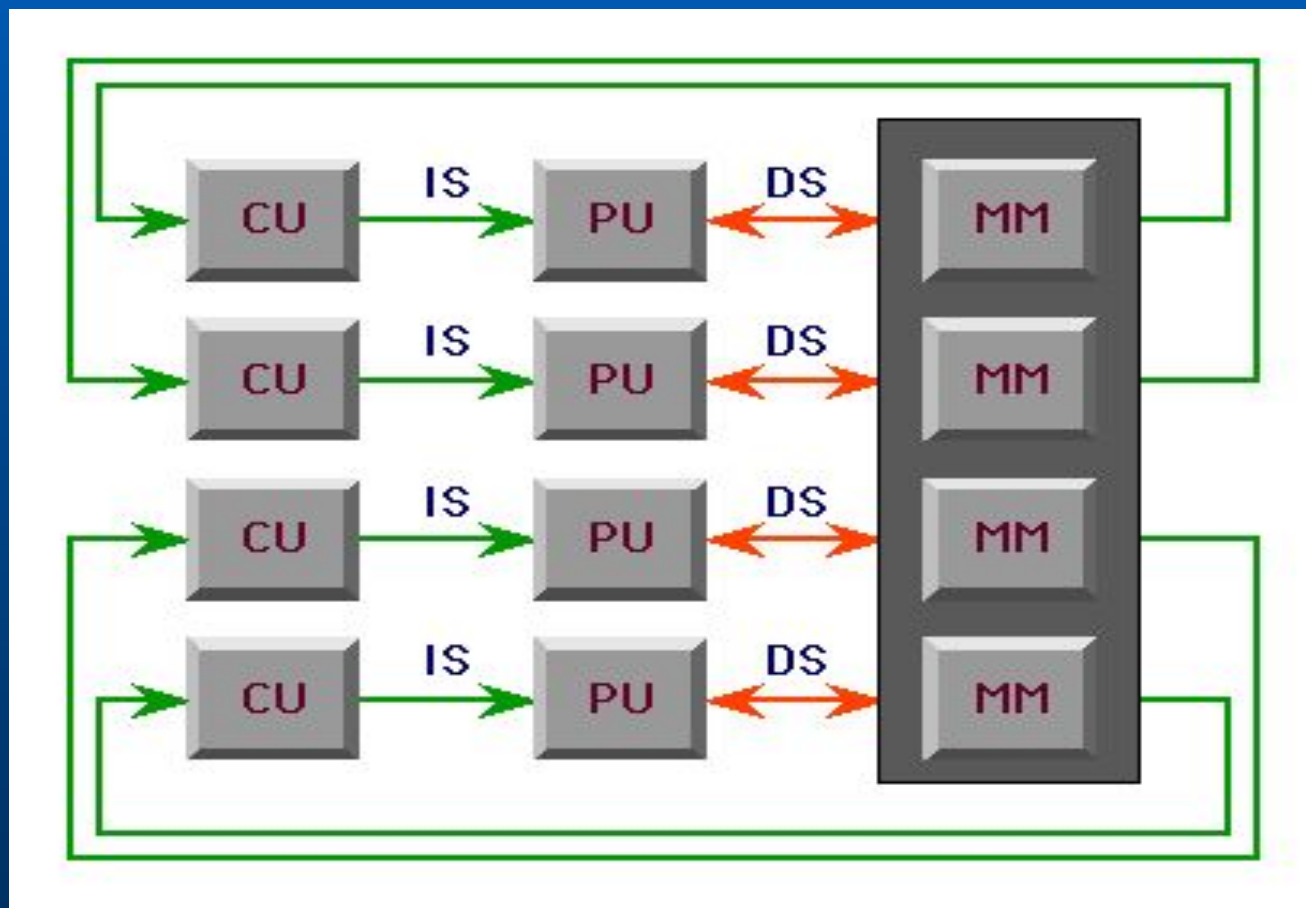
SIMD-компьютеры



MISD-компьютеры



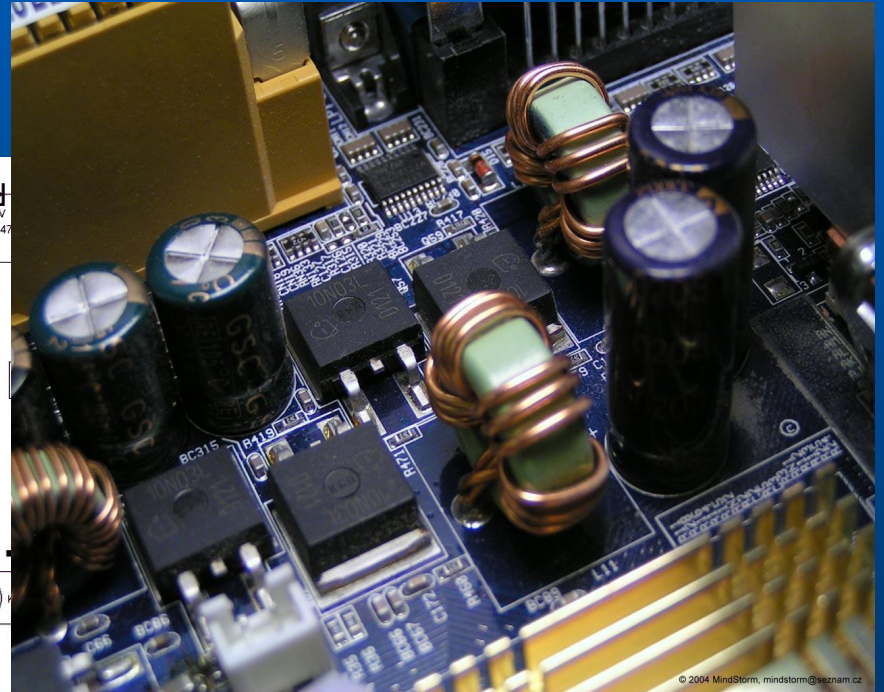
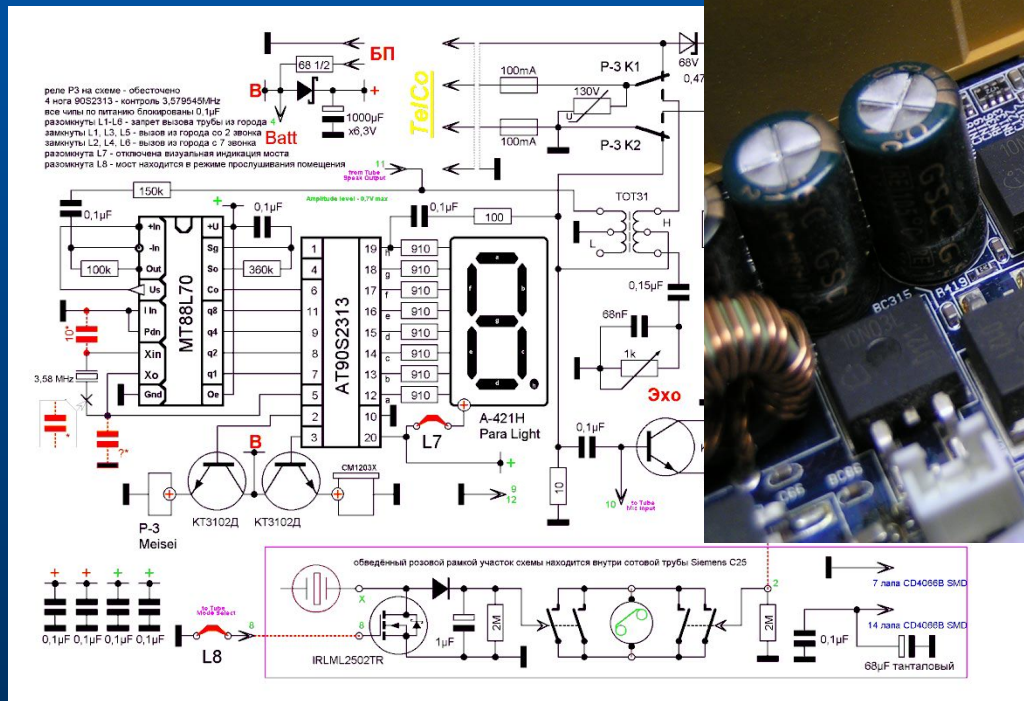
MIMD-компьютеры



Компоненты компьютера

- Центральный процессор и оперативная память
ядро системы
- Внешняя память и устройства ввода/вывода
«периферия»
- Коммуникации между компонентами системы
осуществляется посредством шин

Схема компьютера



© 2004 MindStorm, mindstorm@seznam.cz

In June 2010, 2 systems appeared in the TOP500 list. Rank

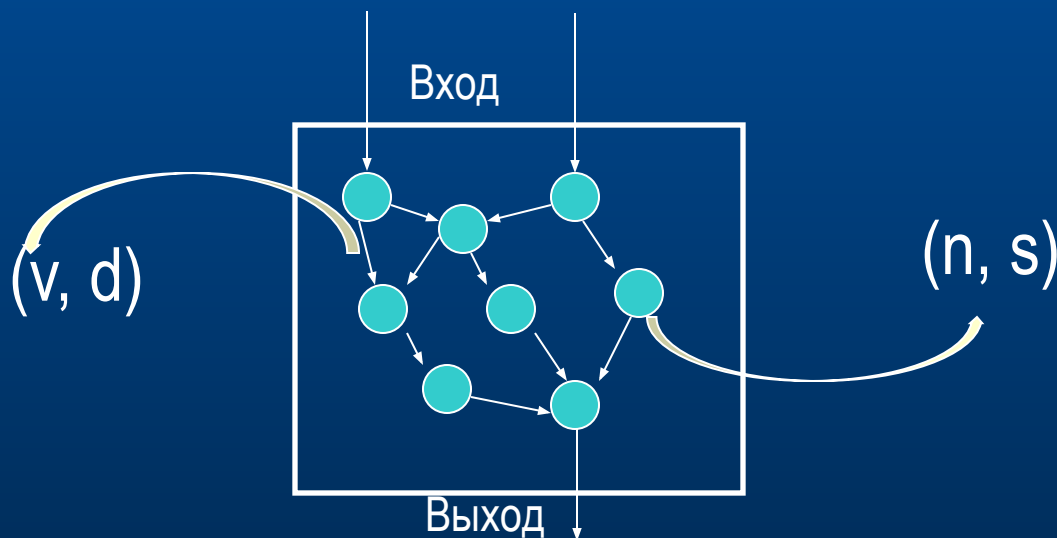
System	Procs	Memory(GB)	Rmax (GFlops)	Rpeak (GFlops)	Vendor
13	<i>Lomonosov T-Platforms T-Blade2, Xeon 5570 2.93 GHz, Infiniband QDR</i>				35360
54312	350100	414419	<i>T-Platforms</i>		
121	<i>SKIF MSUT-Platforms T60, Intel Quadcore 3Ghz, Infiniband DDR 5000</i>				5000 47170
60000	<i>SKIF/T-Platforms</i>				

Технологии параллельного программирования

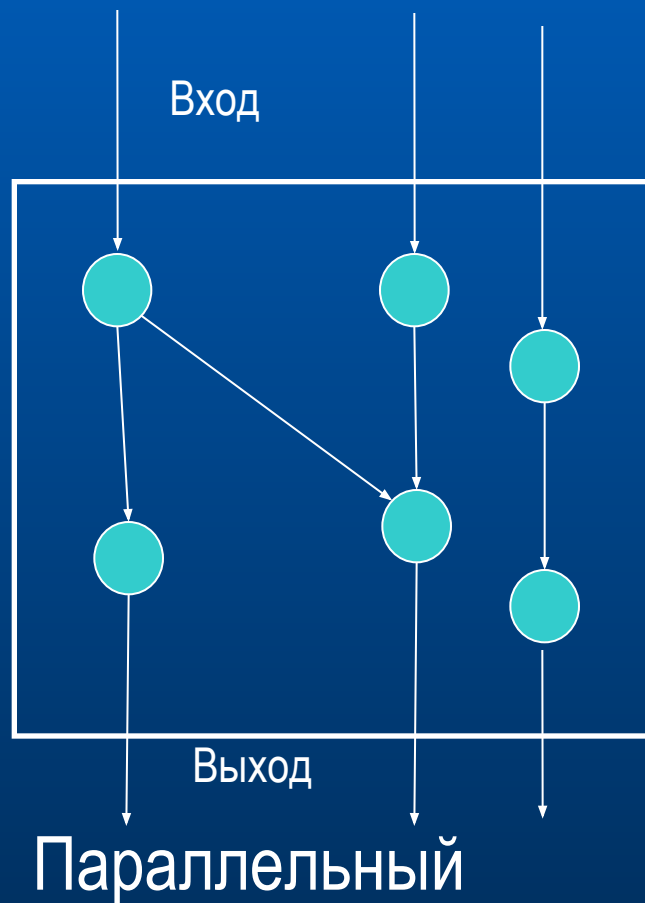
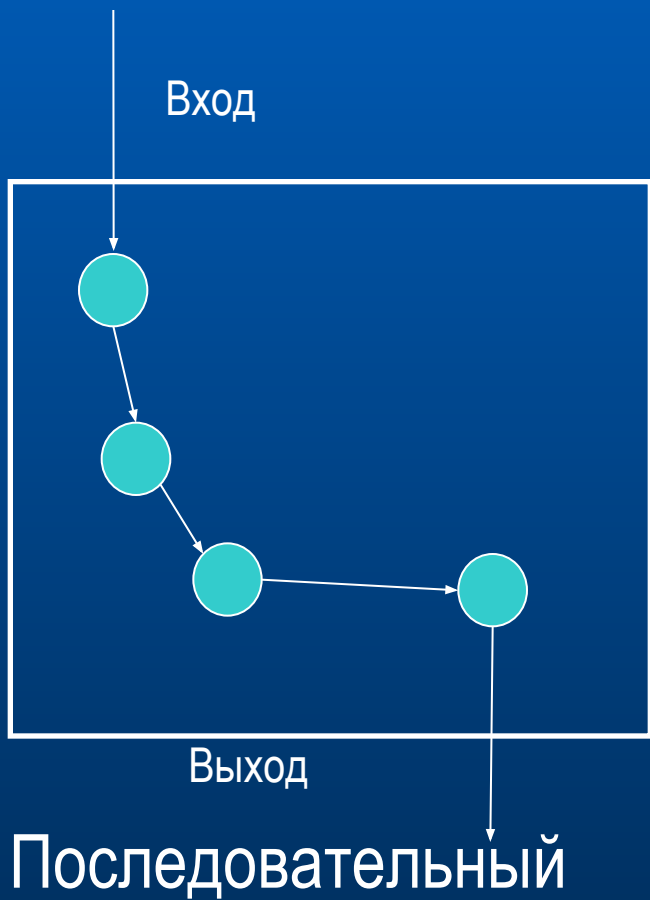
*“Все, что начинается хорошо, кончается плохо.
Все, что начинается плохо, кончается еще хуже.”*
Из законов Мерфи

Особенности программирования параллельных вычислений

- ▶ Модель программирования
 - Алгоритм
 - Информационный граф



Примеры алгоритмов



Основные черты последовательной модели

- Применение стандартных языков программирования
- Хорошая переносимость программ
- Невысокая производительность

Основные черты параллельной модели

- Высокая производительность программ
- Применение специальных приёмов программирования
 - Более высокая трудоёмкость программирования
 - Проблемы с переносимостью программ

Фундаментальные требования к параллельным программам

- Параллелизм
- Локальность
- Масштабируемость
- Детерминизм

Параллелизм

- Данных
- Задач

Параллелизм данных

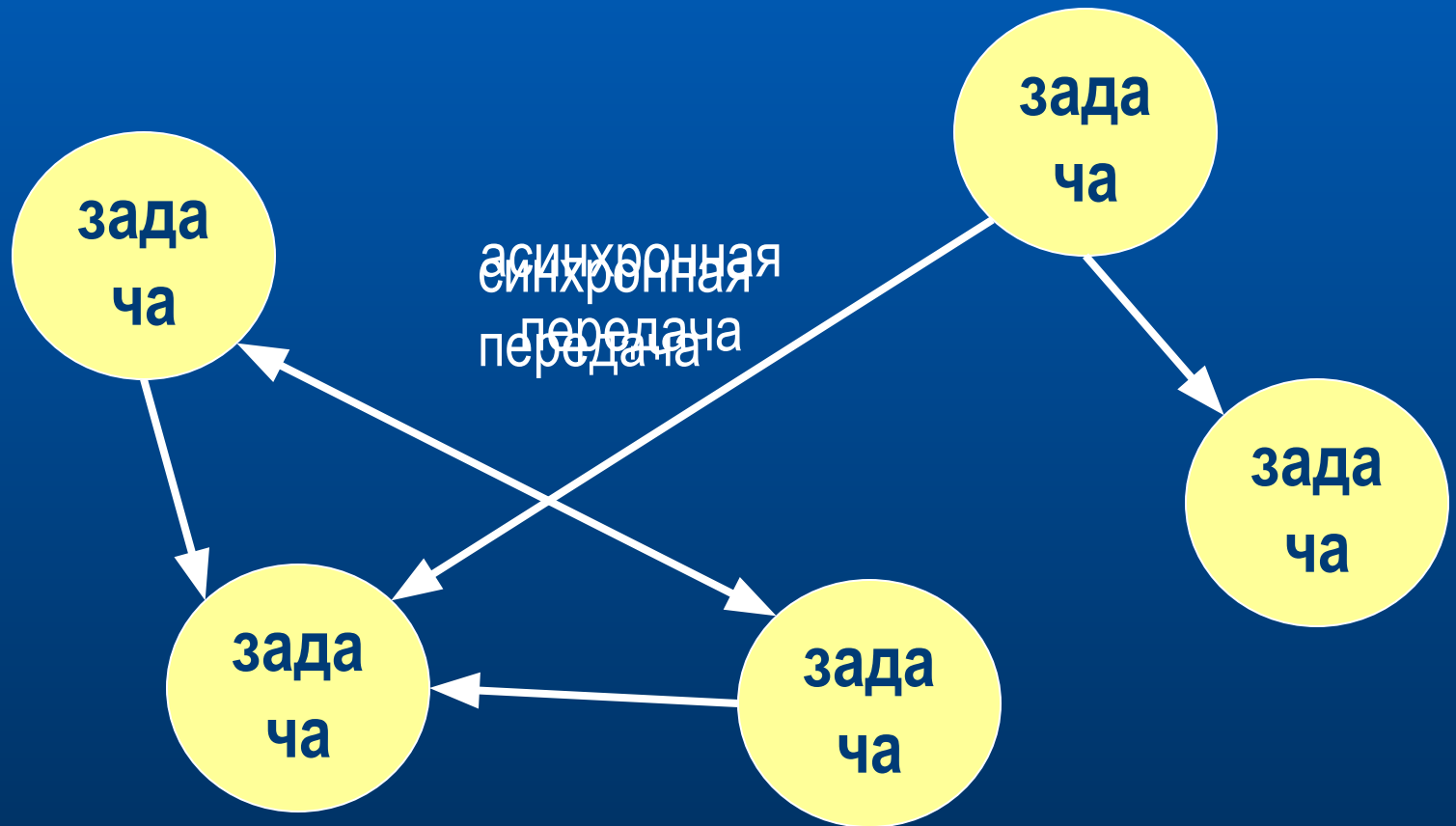
Основные особенности подхода:

- Обработкой данных управляет одна программа
- Пространство имен является глобальным
- Слабая синхронизация вычислений на параллельных процессорах
- Параллельные операции над элементами массива выполняются одновременно на всех доступных процессорах

Параллелизм задач

- Повышенная трудоёмкость разработки и отладки программы
- Программист отвечает за равномерную и сбалансированную загрузку процессоров
- Минимизация обмена данными между задачами
- Возникновение конфликтов

Модель задача/канал



Параллелизм задач

- Большая гибкость
- Большая свобода



**Максимальное
быстроедействие**

Масштабируемость

Независимость результата выполнения параллельной программы от числа процессоров.

Детерминизм



3	+2.688
0	+5.000
1	+1.500
0	+1.125
0	+1.062

Закон Амдала

$$T_1 = T_S + T_P \quad \text{- один процессор}$$

$$T_2 = T_S + T_P / N \quad \text{- } N \text{ процессоров}$$

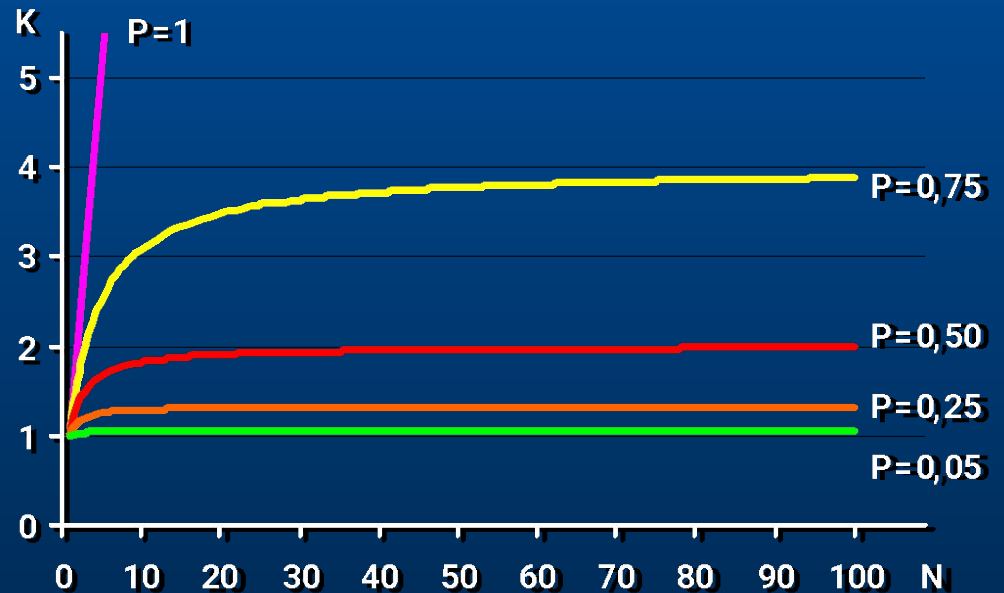
$$K = \frac{T_1}{T_2} \quad \text{- коэффициент ускорения}$$

выполнения программы

Закон Амдала II

- коэффициент ускорения выполнения программы
 - S - доля последовательной части
 - P - доля параллельной части
 - N - число независимых ветвей/процессоров

$$K = \frac{1}{S + \frac{P}{N}}$$



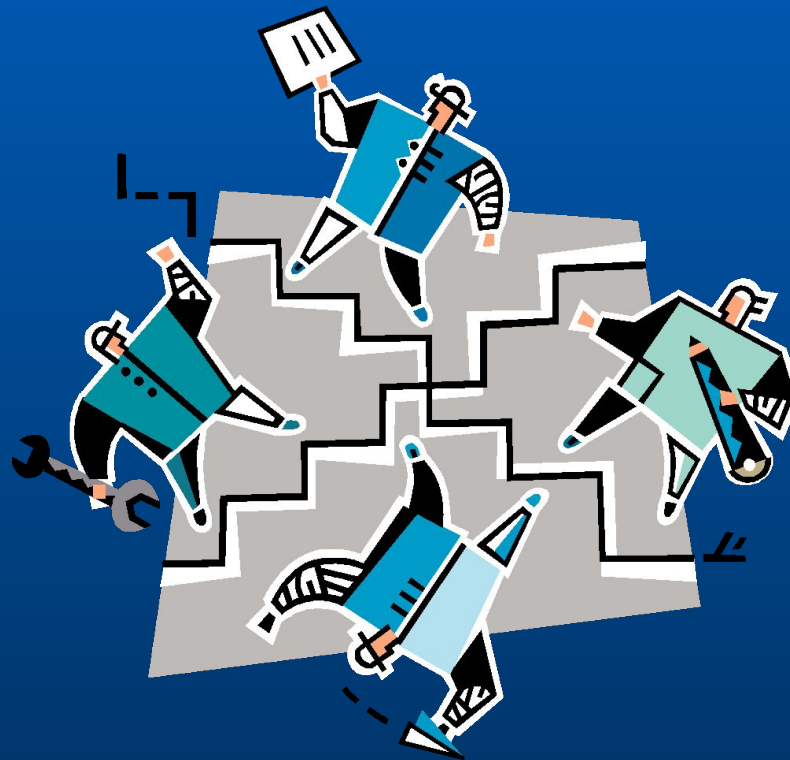
Две парадигмы параллельного программирования

- Равномерная загрузка процессоров
- Эффективная организация обмена информацией между задачами

Базовый набор операций

- Операции управления данными
- Операции над массивами в целом, а также их фрагментами
- Условные операции
- Операции приведения
- Операции сдвига
- Операции сканирования
- Операции, связанные с пересылкой данных

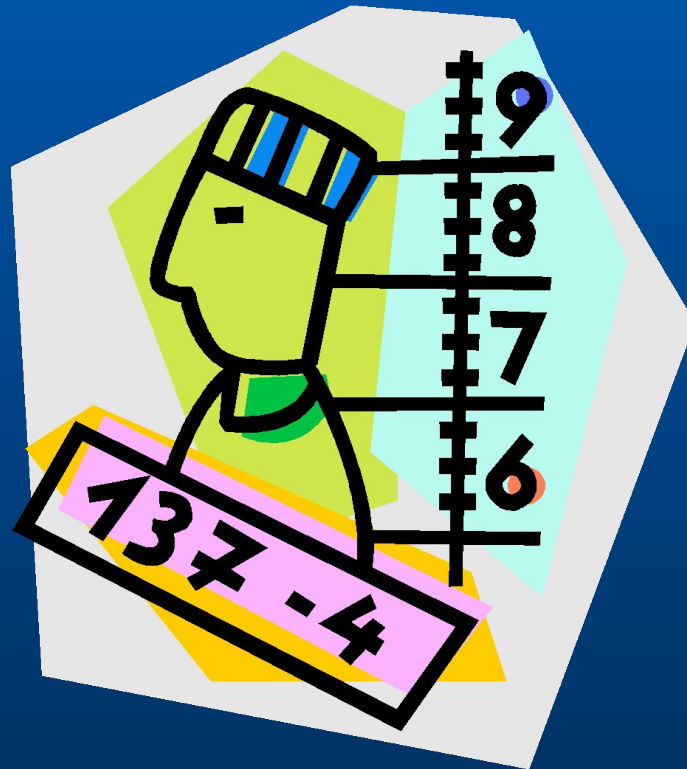
Управление данными



Операции над массивами



Условные операции

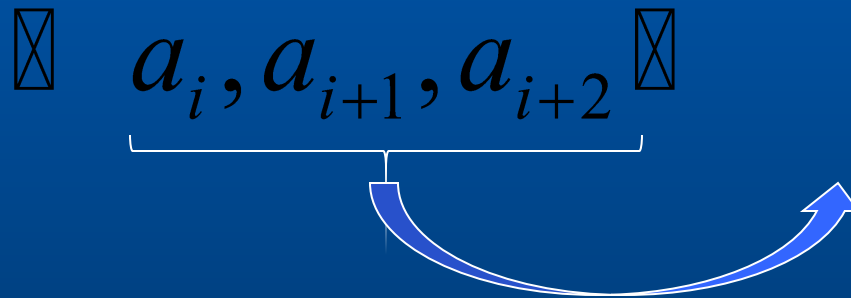


Операции приведения

$$\sum_n a_n$$

$$\max[a_n], i = 1..N$$

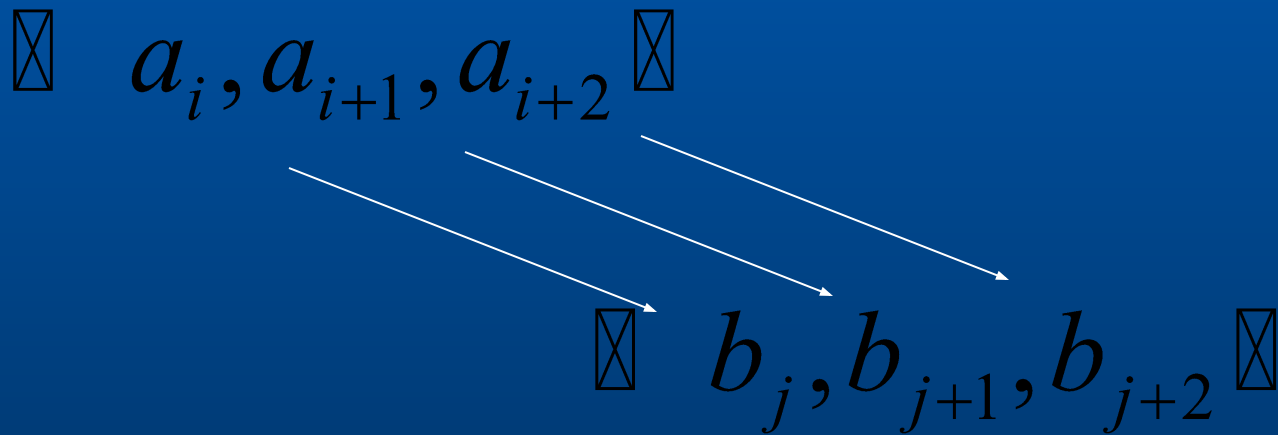
Операции сдвига



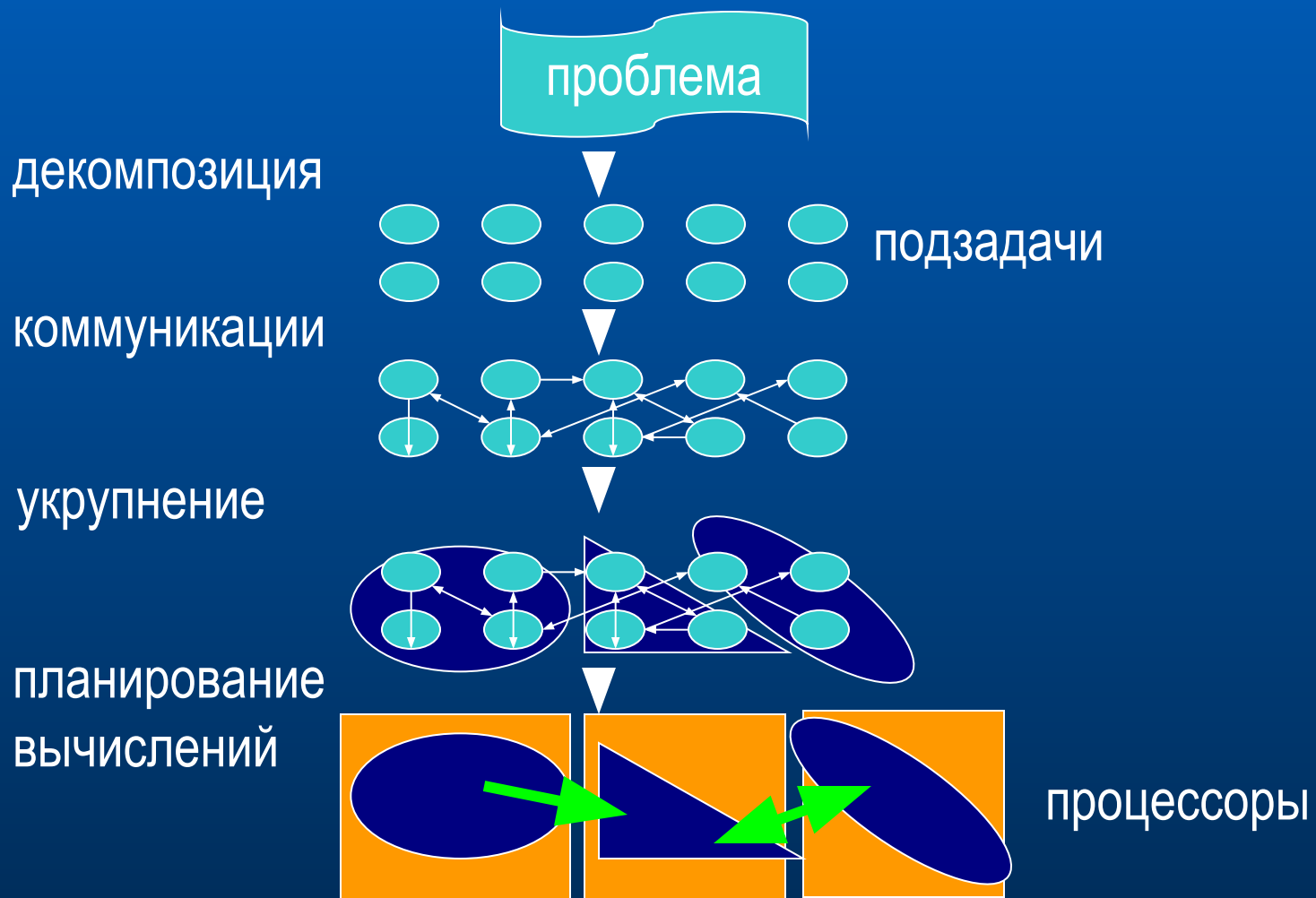
Операции сканирования

$$b_j = \sum_{i=1}^j a_i$$

Операции пересылки данных

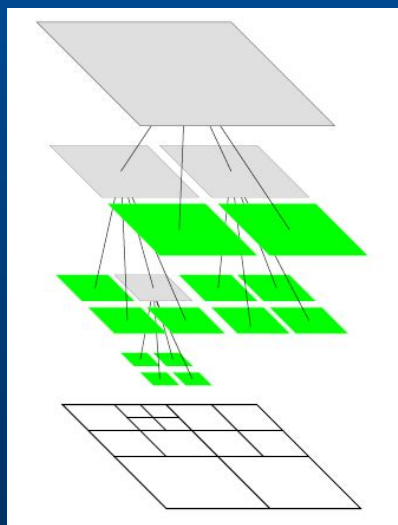


Разработка параллельного алгоритма



Декомпозиция (сегментирование)

- Метод декомпозиции данных
- Метод функциональной декомпозиции



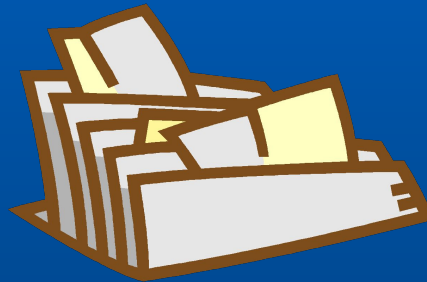
Рекомендации по декомпозиции

- Количество подзадач должно на порядок превосходить количество процессоров
- Следует избегать лишних вычислений и пересылок данных
- Подзадачи должны быть примерно одного размера
- С увеличением объема задачи количество подзадач увеличивалось

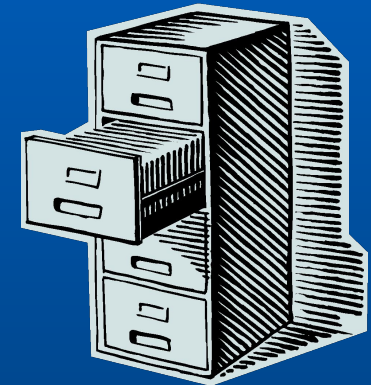
“Зернистость”



мелкозернистая



среднеблочный



крупноблочный

< 20

~ 500

Через разделяемые переменные

независимые
программы

Через сообщения

Независимость частей программы

- Независимость по данным
 - Независимость по управлению
 - Независимость по ресурсам
-

```
do k = 1, m
  if (a[k] < c) then
    a[k] = 1
  endif
enddo
```

```
do k = 1, m
  if (a[k-1] < c) then
    a[k] = 1
  endif
enddo
```

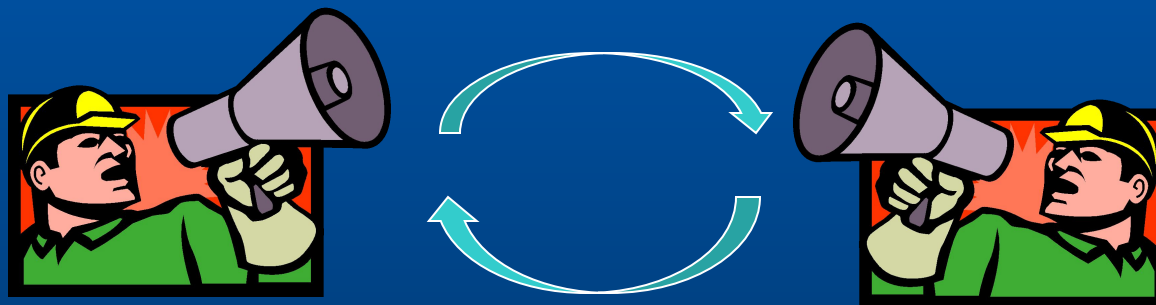

Проектирование коммуникаций

- Локальные
- Глобальные
- Структурированные
- Неструктурированные
- Статические
- Динамические
- Синхронные
- Асинхронные

Рекомендации по проектированию коммуникаций

- Количество коммуникаций у подзадач должно быть примерно одинаковым
- Желательно использовать локальные коммуникации
- Коммуникации должны быть , по возможности, параллельными

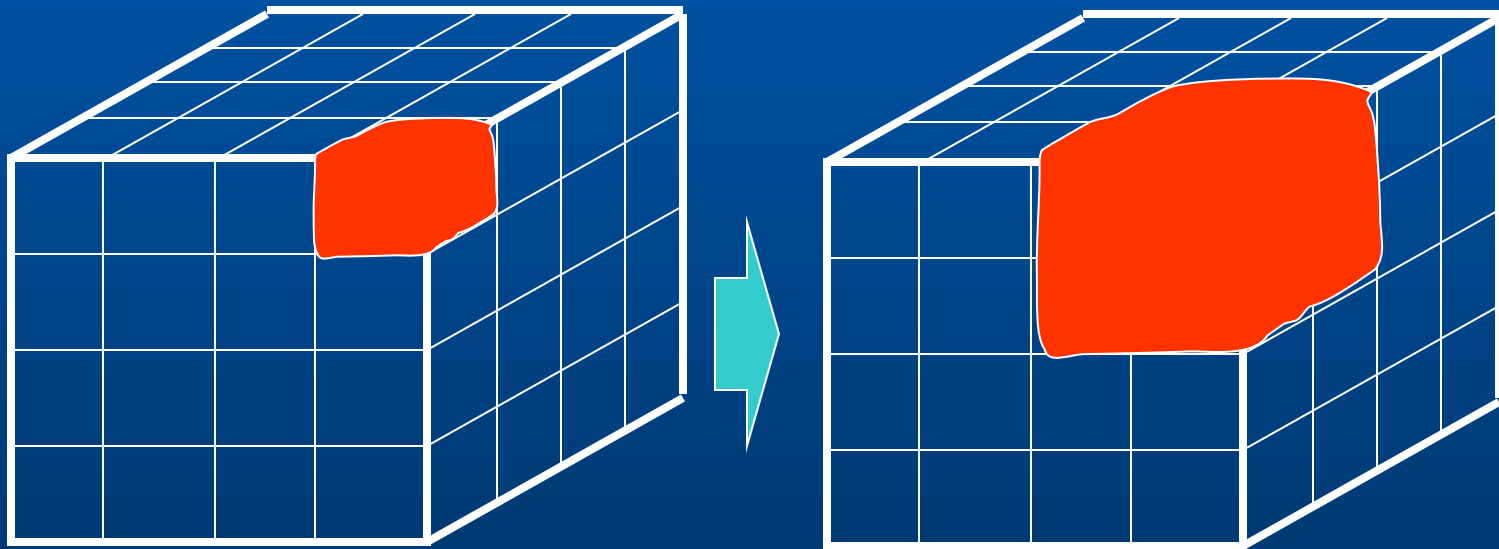
Тупиковые ситуации



Реализация обмена сообщениями

- MPI (Message Passing Interface)
- PVM (Parallel Virtual Machine)
- RPC (Remote Procedure Control)
- CORBA (Common Object Request Broker Architecture)
- DCOM (Distributed Component Object Model)

Укрупнение



Укрупнение

- Снижение затрат на коммуникации
- При необходимом дублировании вычислений или данных не должно быть потерь в производительности или масштабируемости
- Результирующие задачи должны иметь примерно одинаковую трудоёмкость
- Сохранение масштабируемости
- Сохранение возможности параллельного выполнения
- Снижение стоимости и трудоёмкости разработки

Планирование вычислений

- Стратегия управления
хозяин/работник



Характеристики быстродействия

Факторы влияющие на производительность

- Технология выполнения аппаратной части
- Архитектура вычислительной системы
- Методы управления ресурсами
- Эффективность параллельного алгоритма
- Особенности структуры данных
- Эффективность языка программирования
- Квалификация программиста
- Эффективность транслятора

Последовательность действий

- Разработка алгоритма
- Предварительный анализ быстродействия программы
- Оценка трудоёмкости и целесообразности разработки параллельной программы
- **Кодирование**

Список литературы:

- Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем. -СПб.: Петербург, 2002.
- Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. -СПб: BHV, 2002.
- А.С.Антонов "Параллельное программирование с использованием технологии MPI", издательстве Московского университета 2004
- А. А. Букатов, В. Н. Дацюк, А. И. Жегуло. Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Издательство ООО «ЦВВР», 2003, 208 с
- Калиткин Н.Н. Численные методы. - М.: Наука, 1978. - 512 с
- В. Д. Корнеев "Параллельное программирование в MPI", издательство "Регулярная и хаотическая динамика" 2003 г., - 303 стр.
- Грегори Р. Эндрюс "Основы многопоточного, параллельного и распределенного программирования", издательство "Вильямс" 2003 г.- 512 стр.
- Богачев К.Ю. "Основы параллельного программирования" издательство "Бином. Лаборатория знаний" 2003 г. - 342 стр.
- Воеводин В.В. Параллельные структуры алгоритмов и программ. - М.: ОВМ АН СССР, 1987. - 148с.
- <http://parallel.ru>

