1. Введение в дисциплину

1

Классификация вычислительных систем.
Типовые схемы коммуникации в многопроцессорных вычислительных системах.

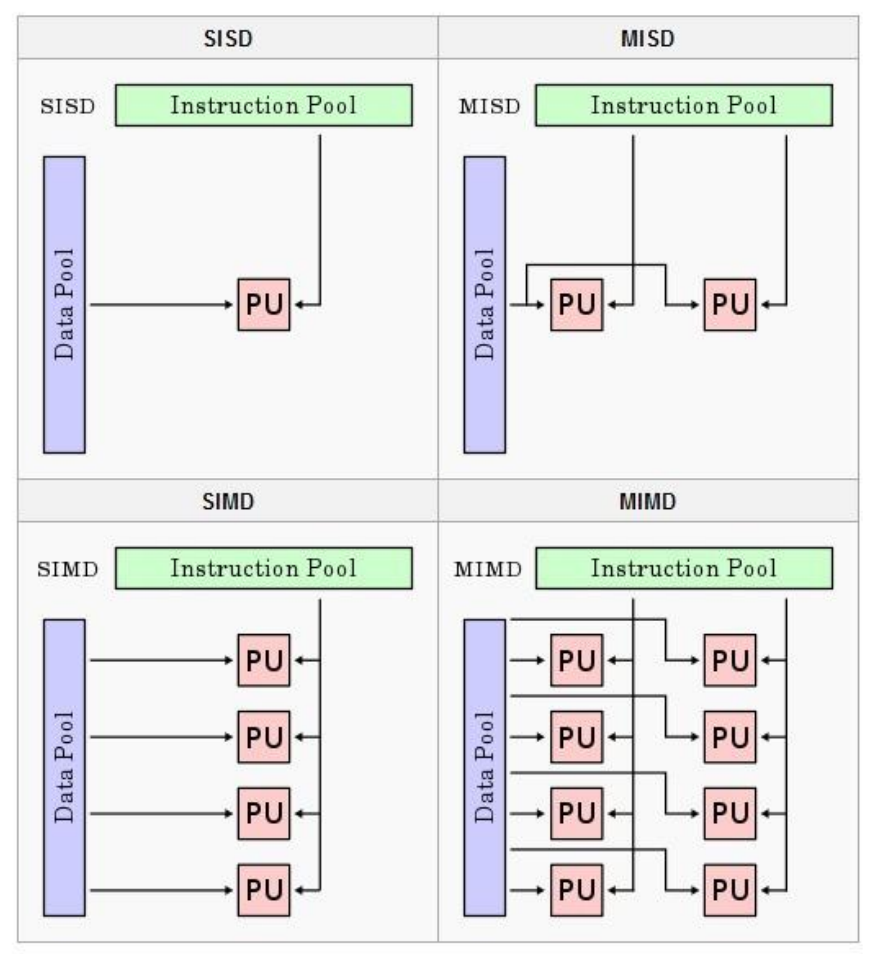
Классификация ПВС по архитектуре

- **Архитектура** ВС - общая логическая организация ВС:
  - определяющая процесс обработки данных,
  - включающая
    - архитектуру ЭВМ,
    - структуру и характеристики программного обеспечения, принципы его взаимодействия с аппаратными средствами.

- **Основа** классификации – систематика Флинна:

  анализ **взаимодействия потоков** выполняемых **команд** и **потоков** обрабатываемых **данных**
  вид *параллелизма* (ПВС)

Основные типы ВС по Флинну
(Michael J. Flynn, Таксономия Флинна - 1966 г.)



Основные типы ВС по Флинну

- **SISD** (Single Instruction Single Data) –
  1 поток команд, 1 поток данных.
  - есть только один поток команд,
  - все команды обрабатываются последовательно друг за другом,
  - каждая команда инициирует одну операцию с одним потоком данных

  П. Стандартный компьютер фон Неймана.

Основные типы ВС по Флинну

- **SIMD** (Single Instruction Multiple Data) –
  1 поток команд, много потоков данных.
  - один поток команд, включающий векторные команды,
  - может выполняться одна операция сразу над многими данными - элементами вектора.

  П1. Компьютер с векторным процессором (операнды – массивы).
  П2. Специализированные многопроцессорные ВС
  (одна команда одновременно выполняется с разными данными)
  для обработки видео, изображений и аудио, для ускорения 3D- и 2D-графики и других мультимедийных задач.

Основные типы ВС по Флинну

- **MISD** (Multiple Instruction Single Data) –
  много потоков команд, 1 поток данных.
  Отказоустойчивые компьютеры,
  ВС с систолическим массивом (*systolic array*) процессоров.
- **MIMD** (Multiple Instruction Multiple Data) –
  много потоков команд, много потоков данных.
  Большинство многопроцессорных ПВС

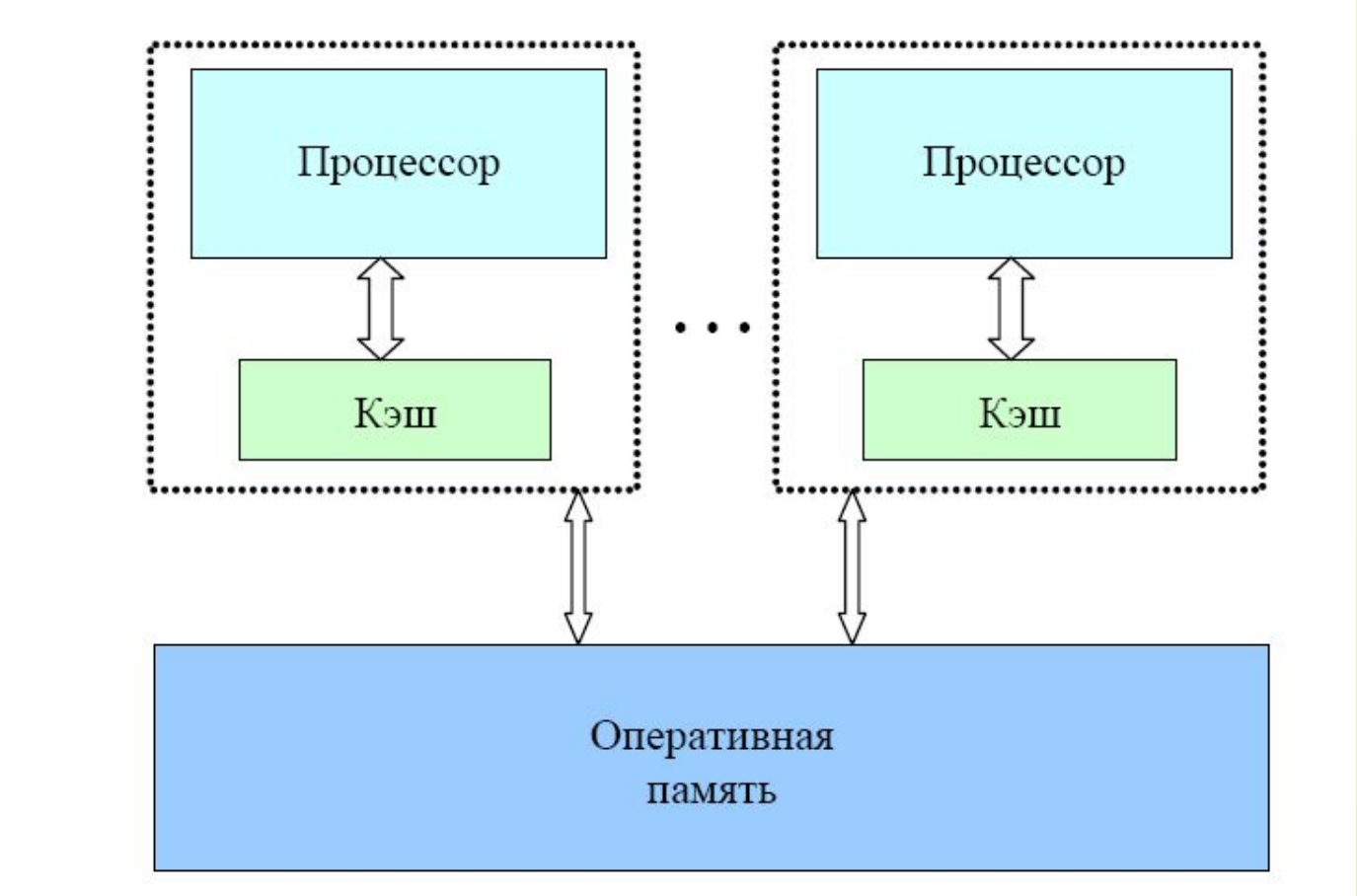
Разновидности ВС типа MIMD

- Дальнейшая классификация ВС – по способам организации оперативной памяти:
  - **Мультипроцессоры** – ВС с **общей, разделяемой** между процессорами памятью.
  - **Мультикомпьютеры** – ВС с **распределенной** памятью самостоятельных компьютеров, объединенных в сеть (MPP-системы, Massively Parallel Processing).

Мультипроцессоры – способы построения общей памяти

- **Единая общая память** с равноправным (однородным) доступом (**Uniform Memory Access, UMA**).
- Используется в ВС на основе:
  - симметричных мультипроцессоров (**SMP-системы**, Symmetric Multiprocessing) П. IBM eServer,
  - векторных параллельных процессоров, в которых предусмотрены команды однотипной обработки векторов независимых данных (**PVP-системы**, Parallel Vector Processor)

Архитектура систем UMA



- ***Cache memory*** – кэш-память, **кэш**: **промежуточный буфер с быстрым доступом**, содержащий информацию, которая может быть запрошена с наибольшей вероятностью.

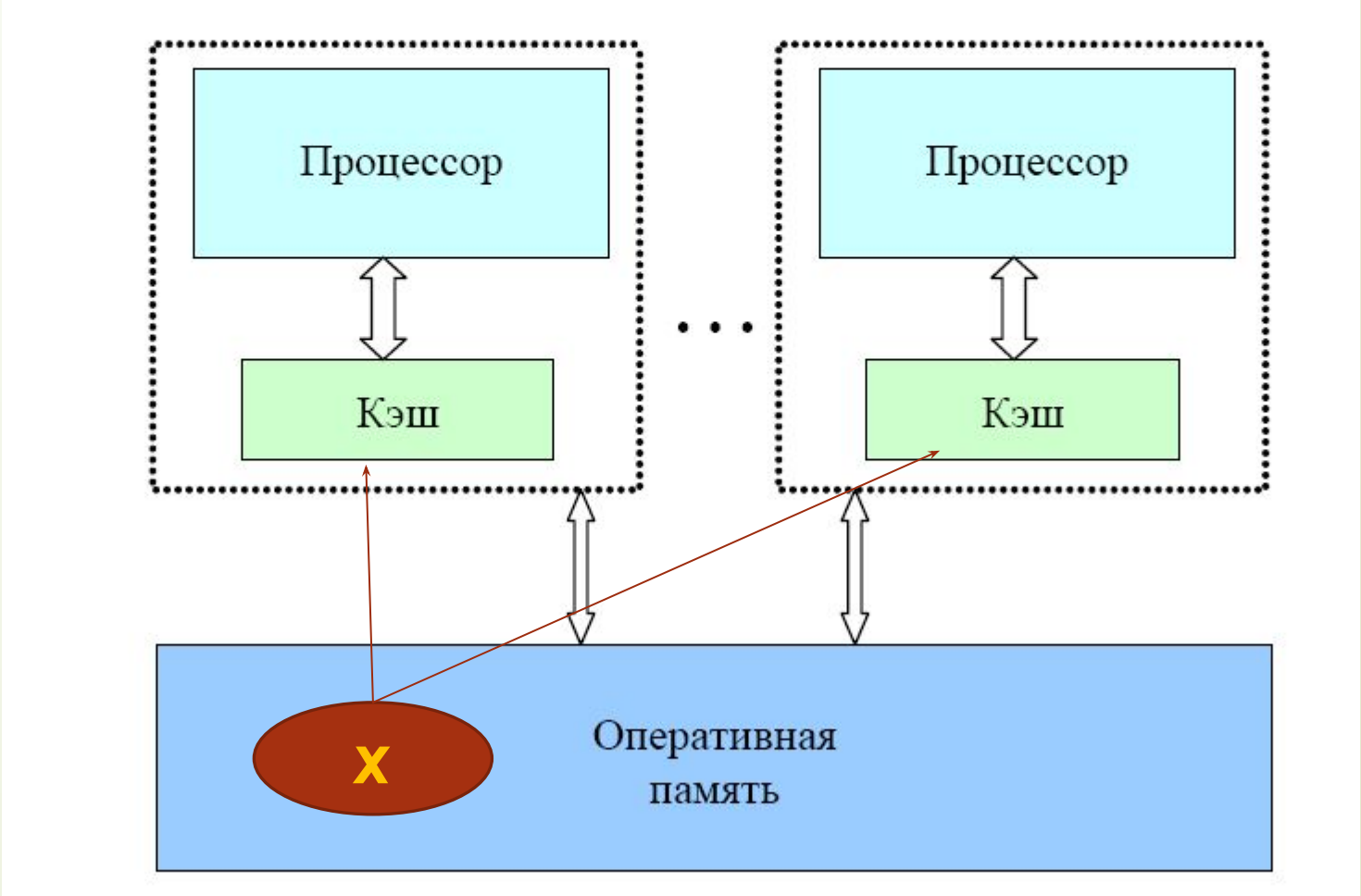
- Доступ к данным в кэше быстрее, чем выборка исходных данных из оперативной памяти
- Уменьшается время доступа к данным
- Увеличивается общая производительность ВС.

- *Кэширование применяется жесткими дисками, браузерами, Web-серверами, службами DNS и т.д.*

Проблемы UMA. *Однозначность данных*

- Доступ с разных процессоров к общим данным
- Необходимо обеспечивать **однозначность** *(когерентность)* содержимого **разных** кэшей *(cache coherence problem)*.
- **Решение**: уведомление всех процессорных узлов (и кэшей!) об изменении значения в общей памяти

Копии значения переменной **x** – в разных кэшах
Возможно изменение **x** одним из процессоров.

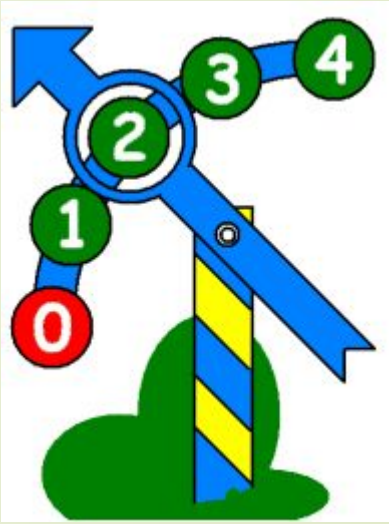


Проблемы UMA. Синхронизация

- Доступ с разных процессоров к общим данным
- Необходимость **синхронизации** взаимодействия одновременно выполняемых **потоков команд**
- **Методы синхронизации:**
  - **Семафор**
  - **Мьютекс**
    (mutex, mutual exclusion, взаимоисключение)
  - **Критические секции**
  - **События**

Синхронизация. Семафор

- **Семафор – системный объект,** с набором методов.
- В C/C++, C# можно работать с семафорами через стандартные классы.
- **Семафор может обеспечить**:
  - запрет одновременного выполнения заданных процессов или потоков;
  - ограничение на число параллельных потоков;
  - поочерёдный доступ к ресурсам, для которых невозможен одновременный доступ.



Синхронизация. Мьютекс

- **Мьютекс – системный объект,** с набором методов.
- В C/C++, C# можно работать с мьютексами через стандартные классы.
- **Мьютекс может находиться в 2 состояниях**:
  - свободен;
  - занят.
- **Мьютекс может обеспечить поочередное выполнение 2 потоков, работающих с общим ресурсом.**
- **Мьютекс по работе = двоичному семафору**

Синхронизация. Критические секции

- **Критическая секция -** участок кода, который может одномоментно выполнять **только один поток**.
- В программах помечается:
  - вход;
  - выход.
- **Критическая секция может обеспечить защищенное изменение глобальных переменных.**

Синхронизация. События

- **Событие -** объект, который может быть в состоянии **нейтральном или сигнализирующем**.
- Поток может ждать сигнала о событии для начала выполнения.
- **Возможно взаимооповещение потоков (процессов) о событиях.**

Рисунки из статьи

http://www.viva64.com/ru/a/0037/

http://www.osp.ru/pcworld/2007/10/4623726/

Игорь Орещенков

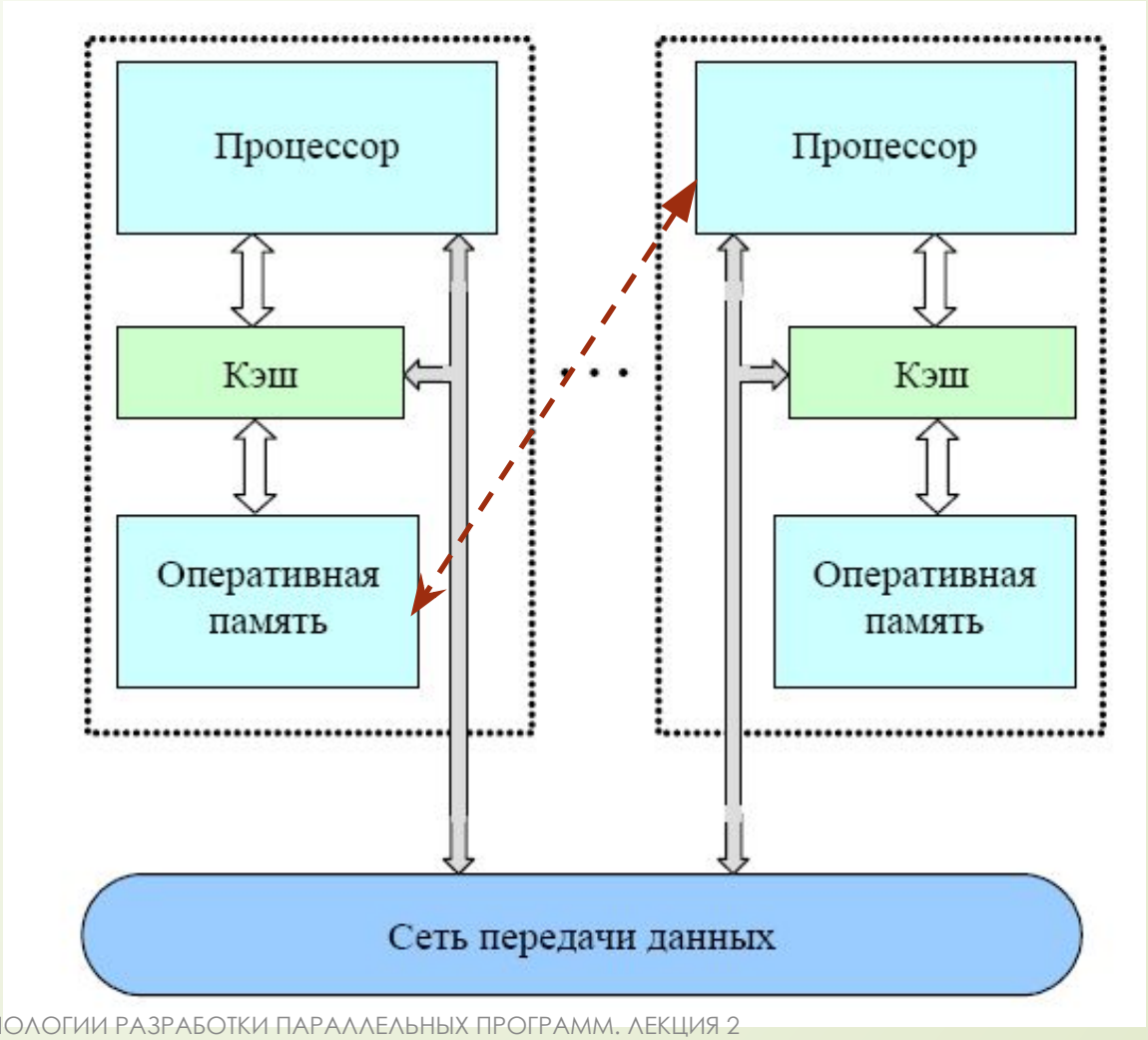
Мультипроцессоры – способы построения общей памяти

2. **Физически распределенная общая память** с неравноправным (неоднородным) доступом (**Non-Uniform Memory Access, NUMA**).

   Принцип:

   - Блок памяти ↔ процессор (быстрый доступ)
   - «Чужие» блоки ↔ процессор (медленный доступ – до нескольких порядков) – **ОСНОВНАЯ ПРОБЛЕМА**!

Архитектура систем NUMA



NUMA системы

- Для данных используются только локальные кэши процессоров – нет общей памяти => нет проблемы когерентности
  (**COMA**-системы, *Cache-Only Memory Architecture*)
- Обеспечена когерентность локальных кэшей (аппаратно)
  (**CC-NUMA-**системы, *Cache-Coherent*)
- Не обеспечена когерентность локальных кэшей
  (**NCC-NUMA-**системы, *Non-Cache-Coherent*)

Мультикомпьютеры

- МК – ВС с **распределенной** памятью самостоятельных компьютеров, объединенных в сеть
- МК – системы типа **NORMA**
  (No-Remote Memory Access – «нет доступа к удаленной памяти»)
- Архитектура **аналогична**
  архитектуре МП с распределенной памятью.
- **НО!**:
  - Каждый процессор может **использовать только свою** локальную память.
  - Для доступа к «чужим» данных д.б. **явно выполнены операции передачи сообщений** (*message passing operations*) – 1- или 2-сторонний обмен данными

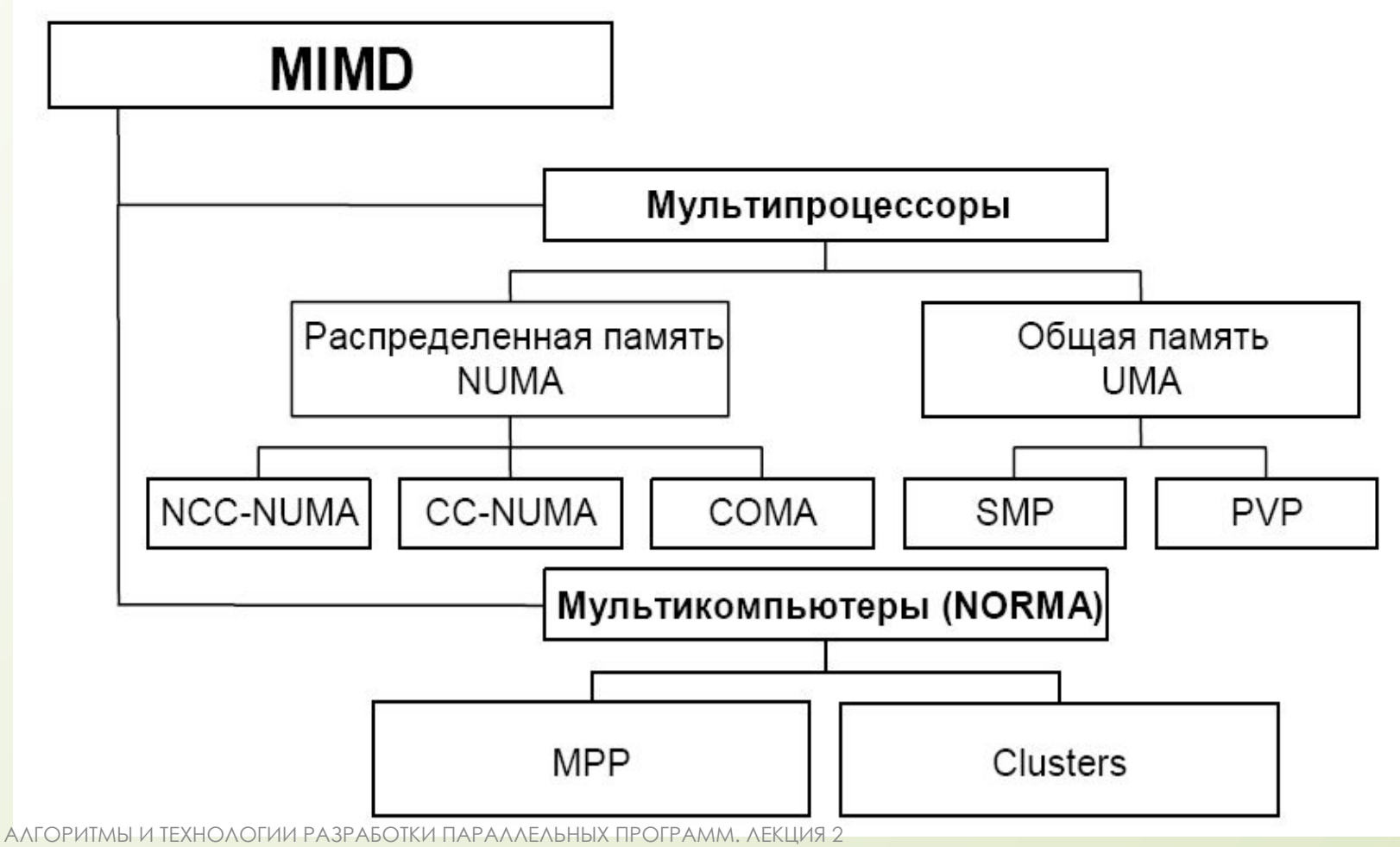
Основные типы МК – многопроцессорных вычислительных систем

- Массивно (*массово*)-параллельные системы, **MPP-**системы (*Massively Parallel Processing* – массово-параллельная обработка)
- Система состоит из однородных *узлов* *(до 10^3)*, включающих:
  - один или несколько ЦП (обычно RISC),
  - локальную память (прямой доступ к памяти других узлов невозможен!),
  - коммуникационный процессор или сетевой адаптер
  - жесткие диски и/или другие устройства I/O

Основные типы МК – многопроцессорных вычислительных систем

- **Кластер** - набор рабочих станций (или даже ПК) общего назначения, используется в качестве дешевого варианта MPP-системы.
- **Связь** узлов - одна из стандартных сетевых технологий (Fast/Gigabit Ethernet, Myrinet и др.) на базе шинной архитектуры или коммутатора.

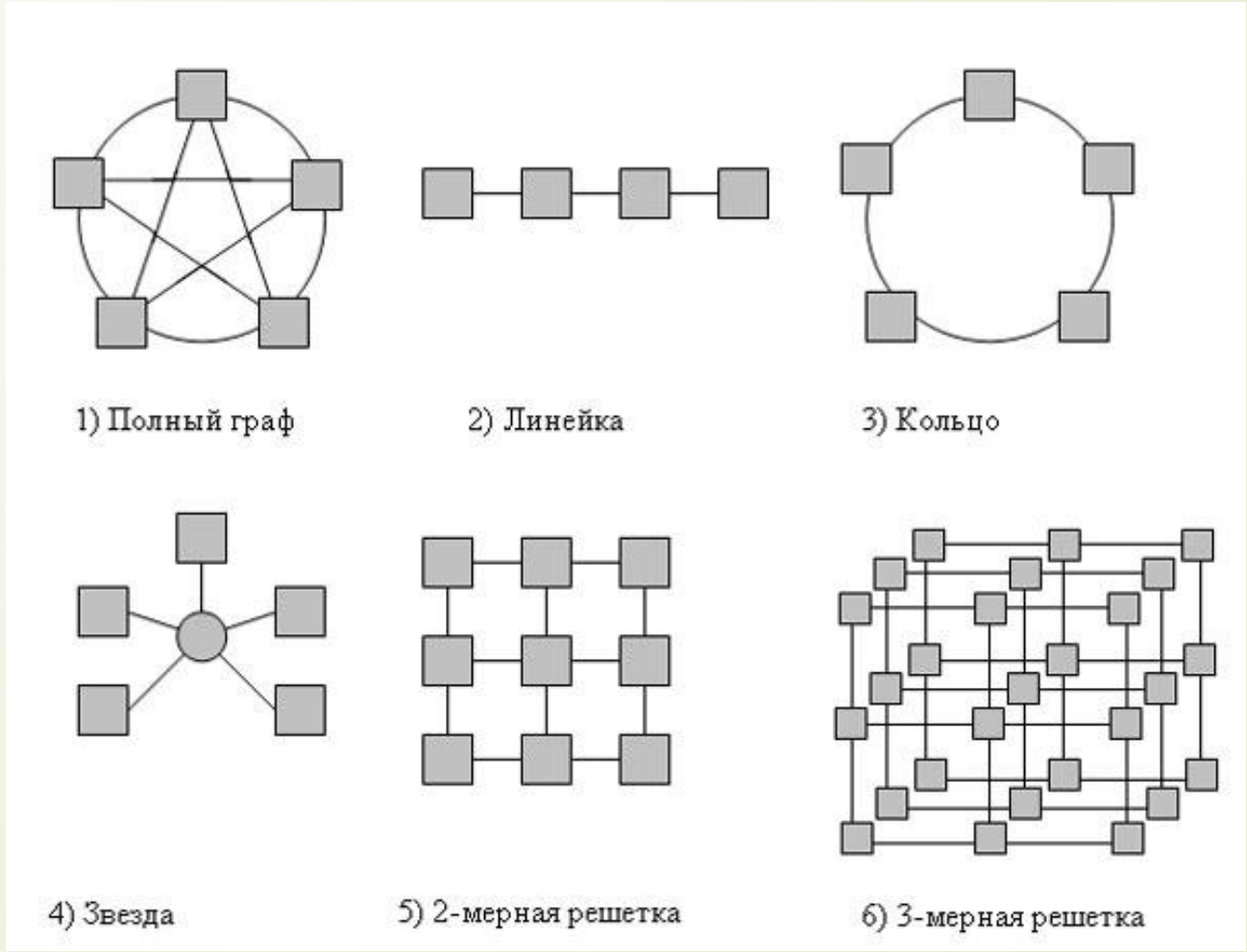
Классификация многопроцессорных ВС (подробно см. http://parallel.ru/computers/classes.html)



Коммуникация в МВС

- **Коммуникация** между процессорами **обеспечивает**:
  - взаимодействие, синхронизацию, взаимоисключения выполняемых процессов
- **Коммуникационная «трудоемкость» алгоритма влияет** на выбор **способа решения** задачи
- Коммуникация определяется **топологией** сети – **схемой расположения и соединения сетевых устройств.**

Примеры топологий сетей в МВС



Особенности топологий сетей передачи данных

Тип - Преимущества - Реализация

- **Полный граф** – **минимум затрат на передачу данных** - **-**
  (кластер с соединением CPU через свитч с ограничением:
  только одна одномоментная операция приема-передачи данных для каждого процессора взаимодействующие пары CPU не должны пересекаться).
- **Линейка** – **идеально для конвейерных вычислений** - **+**
- **Кольцо** - **?** - **+**
- **Звезда** – **для централизованных схем вычислений** - **+**
- **Решетка** (2D, 3D) - **+** - **+**

Характеристики топологий сети

- **Диаметр** – определяет время передачи данных через **max расстояние** между 2 CPU сети (*расстояние равно величине кратчайшего пути*).
- **Связность** – определяет **наличие разных маршрутов** передачи данных между CPU, min число дуг графа, которые надо удалить для получения 2 несвязных областей.
- **Ширина бисекции** – связность, но 2 области д.б. одинакового размера
- **Стоимость** – общее число линий передачи данных в МВС

Характеристики топологий (p – количество процессоров)

Топология	Диаметр	Ширина бисекции	Связность	Стоимость
Граф	1	$p^2/4$	p - 1	p(p - 1)/2
Звезда	2	1	1	p - 1
Линейка	p - 1	1	1	p - 1
Кольцо	p/2	2	2	p
Решетка (2D)	$2(\sqrt{p} - 1)$	$\sqrt{p}$	2	$2(p - \sqrt{p})$