

Топология вычислительной системы и топология вычислительного процесса

ЦКП ДВВР
ИАПУ ДВО РАН
2011г

Понятие топологии

- Тополо́гия — раздел математики, изучающий в самом общем виде явление непрерывности, в частности свойства пространства, которые остаются неизменными при непрерывных деформациях, например, связность, ориентируемость.
- Когда топология еще только зарождалась (конец XIX века), ее называли геометрией размещения или анализ размещения.

Предмет лекции

- Cache memory
- SMP - Symmetric multiprocessing (UMA)
- NUMA - Non-Uniform Memory Access
- RDMA - remote direct memory access
- Network – общие представления о сетевой топологии
- Ethernet - некоторые особенности Ethernet
- Myrinet – специфика Myrinet
- infiniband fabric – современная коммуникационная среда
- Calculations management - управление процессом вычислений
- pure parallel – недостижимый идеал
- master-slave, client-server, producer-consumer (2 типа процессов)
- fixed topology (grid,tor,ncube) – (однородные процессы)

Взаимодействие на общей памяти vs. передача сообщений

- Общая память, средства контроля:
 - Семафоры
 - Мьютексы
 - Критические секции
- Способы передачи сообщений:
 - IPC messages
 - MPI Messages
 - Socket Messaging

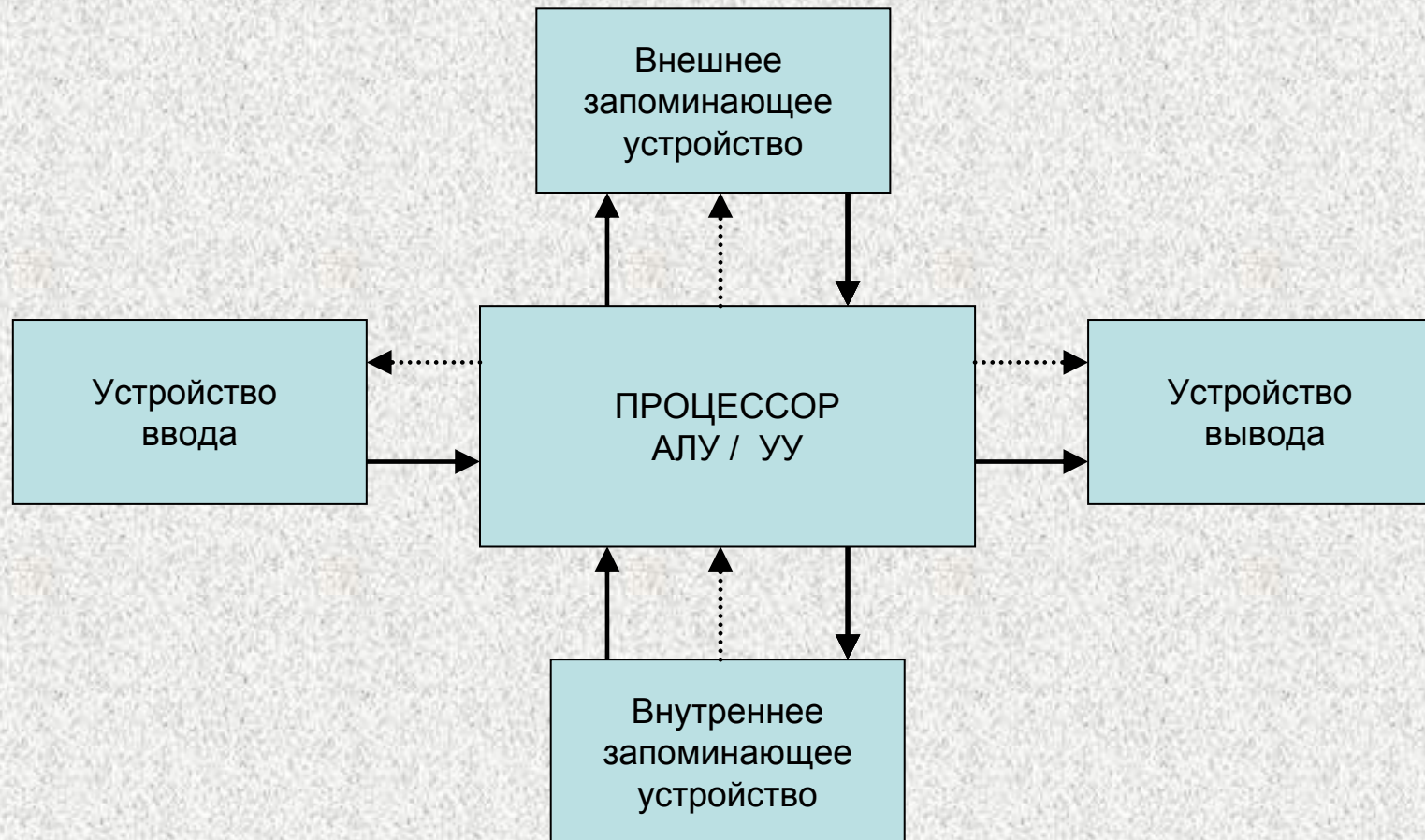
Взаимодействие на общей памяти vs. передача сообщений

- Общая память, основные проблемы:
 - RACE CONDITION
 - Cache coherence
- Основные проблемы при передаче сообщений:
 - DeadLocks (взаимная блокировка процессов)
 - Затраты на синхронизацию процессов

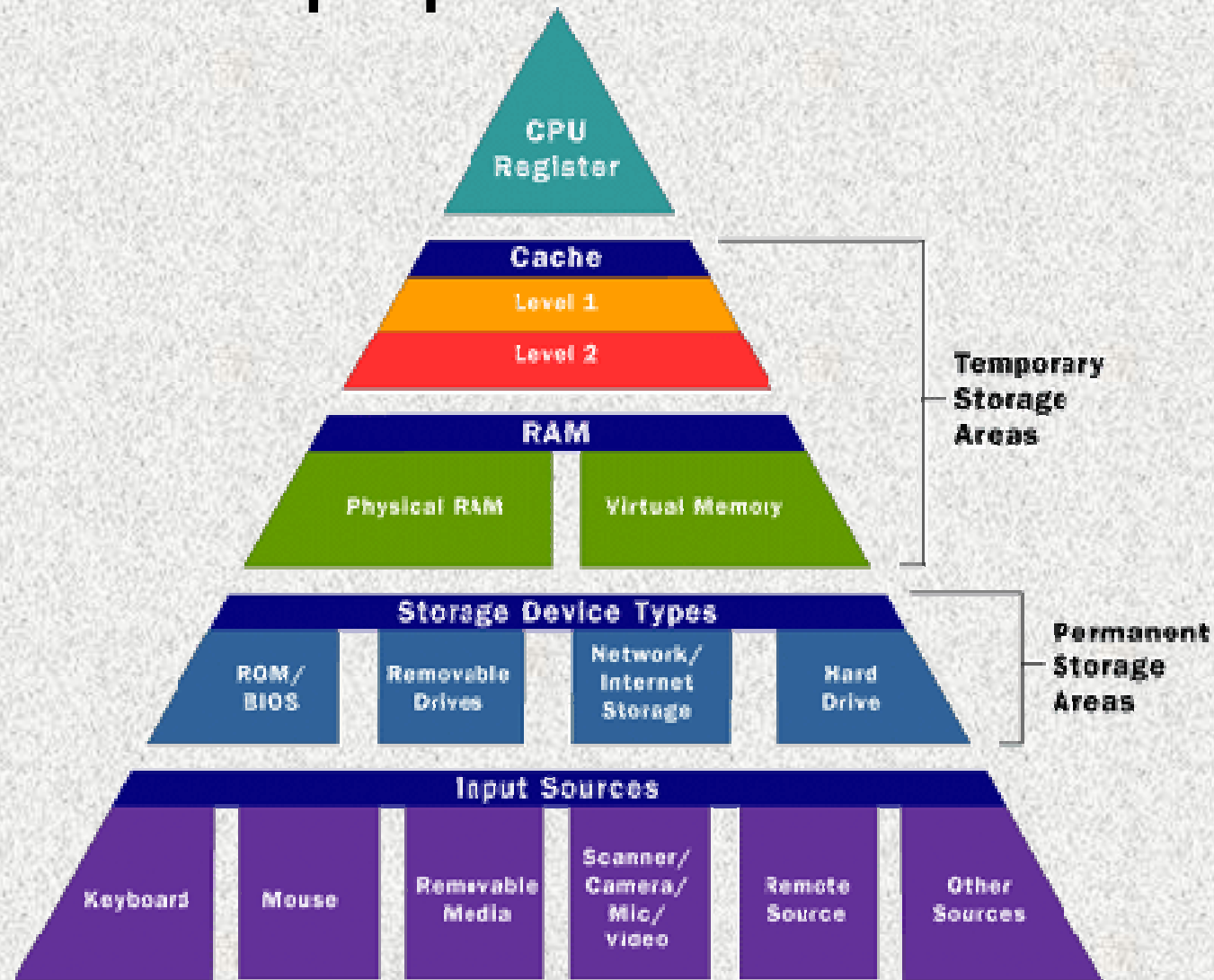
Архитектура Фон-Неймана

- **Архитектура фон Неймана** (англ. von Neumann architecture) — принцип совместного хранения программ и данных в памяти компьютера. (1946г.)
- **Гарвардская архитектура** — архитектура ЭВМ, отличительным признаком которой является раздельное хранение и обработка команд и данных. (Архитектура была разработана Говардом Эйкеном в конце 1930-х годов в Гарвардском университете.)

Архитектура Фон-Неймана



Иерархия памяти



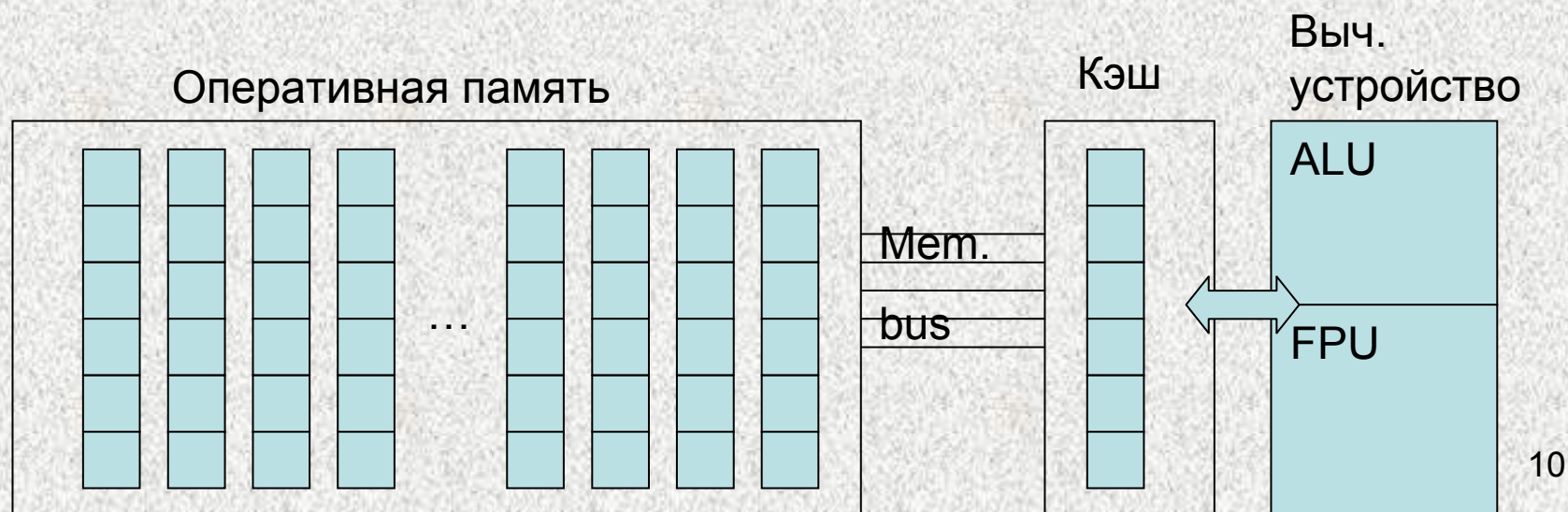
Логическая организация памяти

- Языки «С», «С++» - указатели и ссылки
- Языки Java, С# - ссылки (нет прямого доступа к памяти)
- Представление памяти – прямая лента.



Физическая организация памяти

- MMU – memory management unit
- Mapped memory
- DMA (direct memory access)
- Представление памяти – memory bank, rank, memory bus, cache, page.

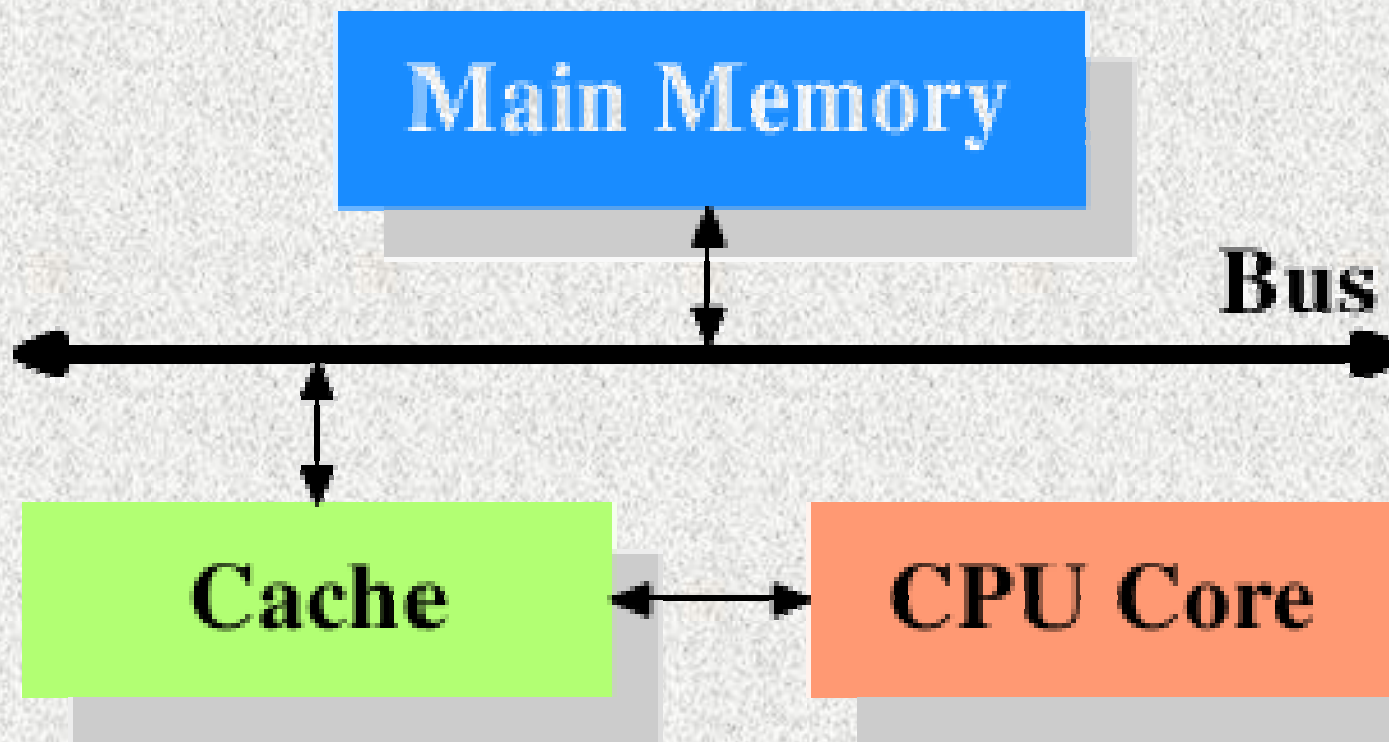


Cache

- Кэш — это память с большей скоростью доступа, предназначенная для ускорения обращения к данным, содержащимся постоянно в памяти с меньшей скоростью доступа
- Кэш память ассоциативна. Каждая запись кэш ассоциирована с блоком данных, который является копией блока в основной памяти
- В современном компьютере процессор имеет непосредственное дело только с кэш памятью.

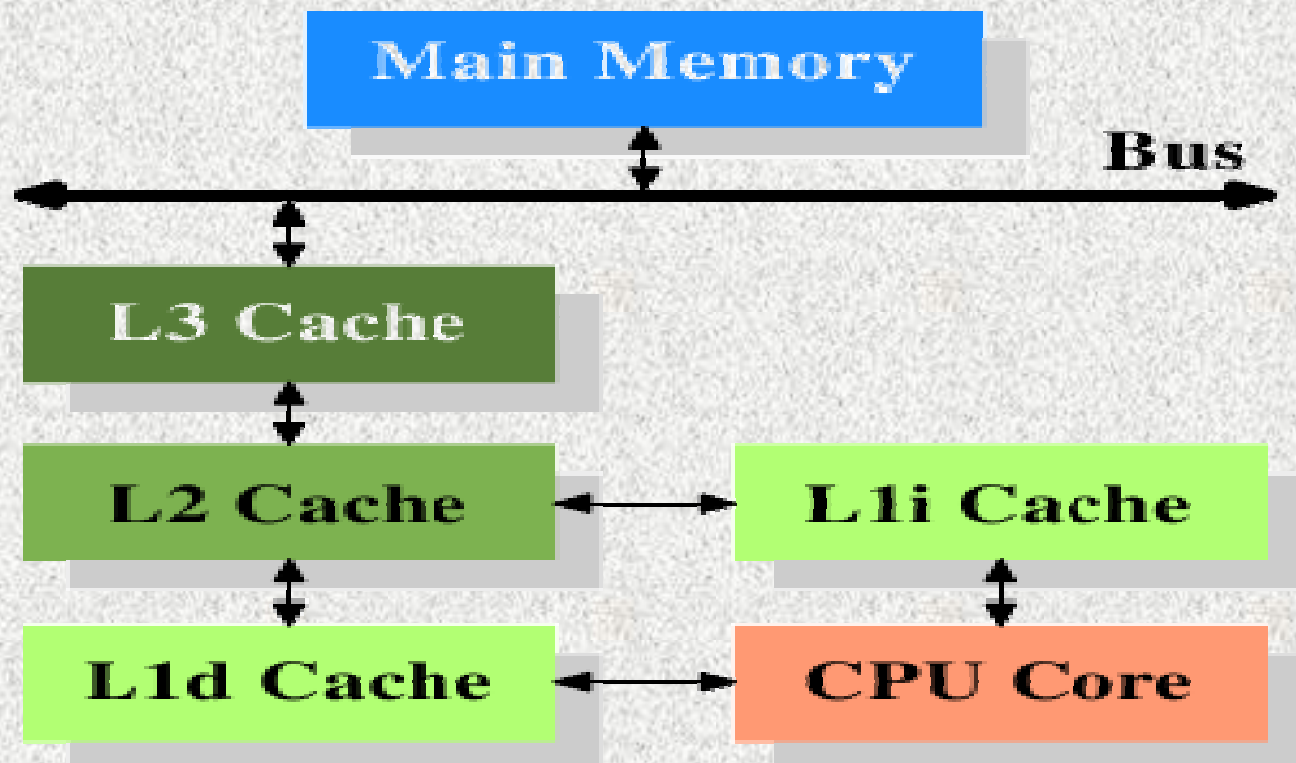
Cache

- *cache miss* - ситуация, когда процессору необходимы данные отсутствующие в кэш (прерывание для загрузки страницы памяти в кэш)
- *hit rate (hit ratio)* – процент успешного обращения к кэш



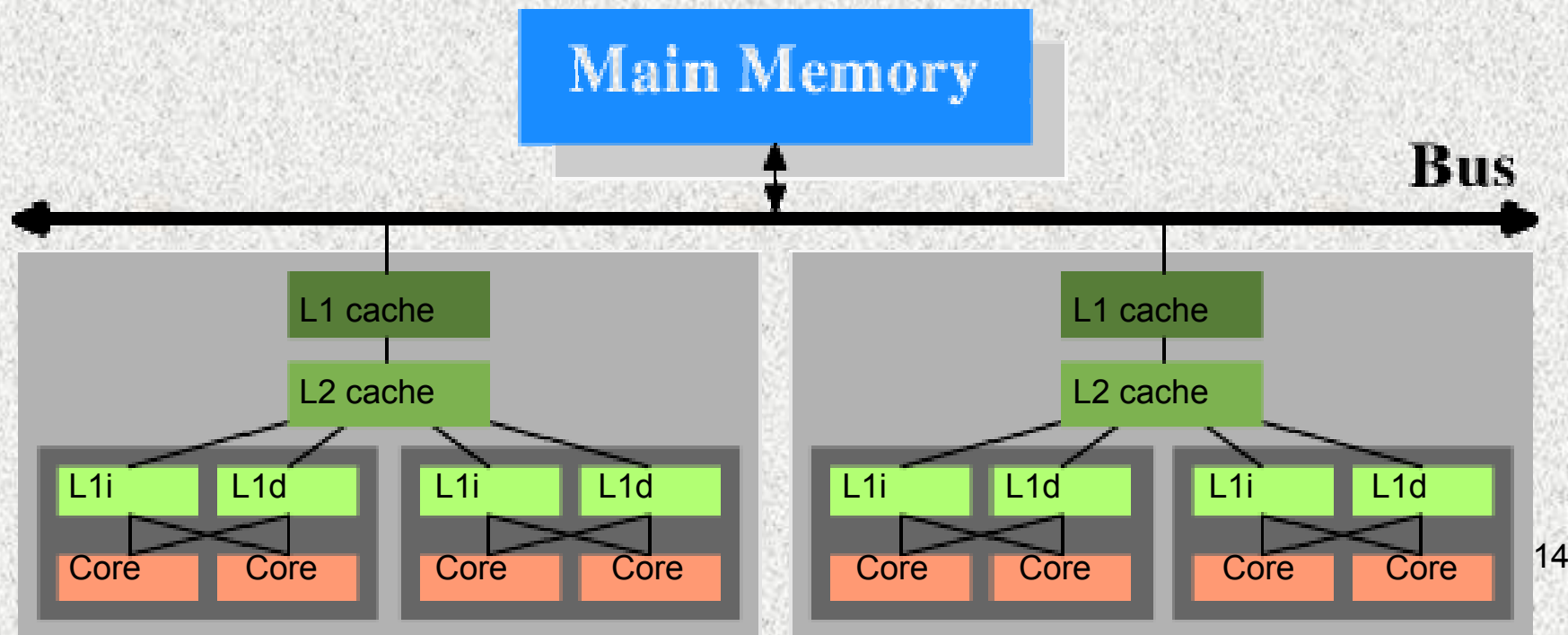
Cache

- В современных процессорах имеется несколько уровней кэш, а также деление на кэш команд и кэш данных.
- при промахе в кэш может быть принято решение отбросить некоторую запись для освобождения пространства. Для выбора отбрасываемой записи используются разные алгоритмы вытеснения.



Cache

- Протоколы взаимодействия между кэшами, которые сохраняют согласованность данных, называют протоколами когерентности кэша.

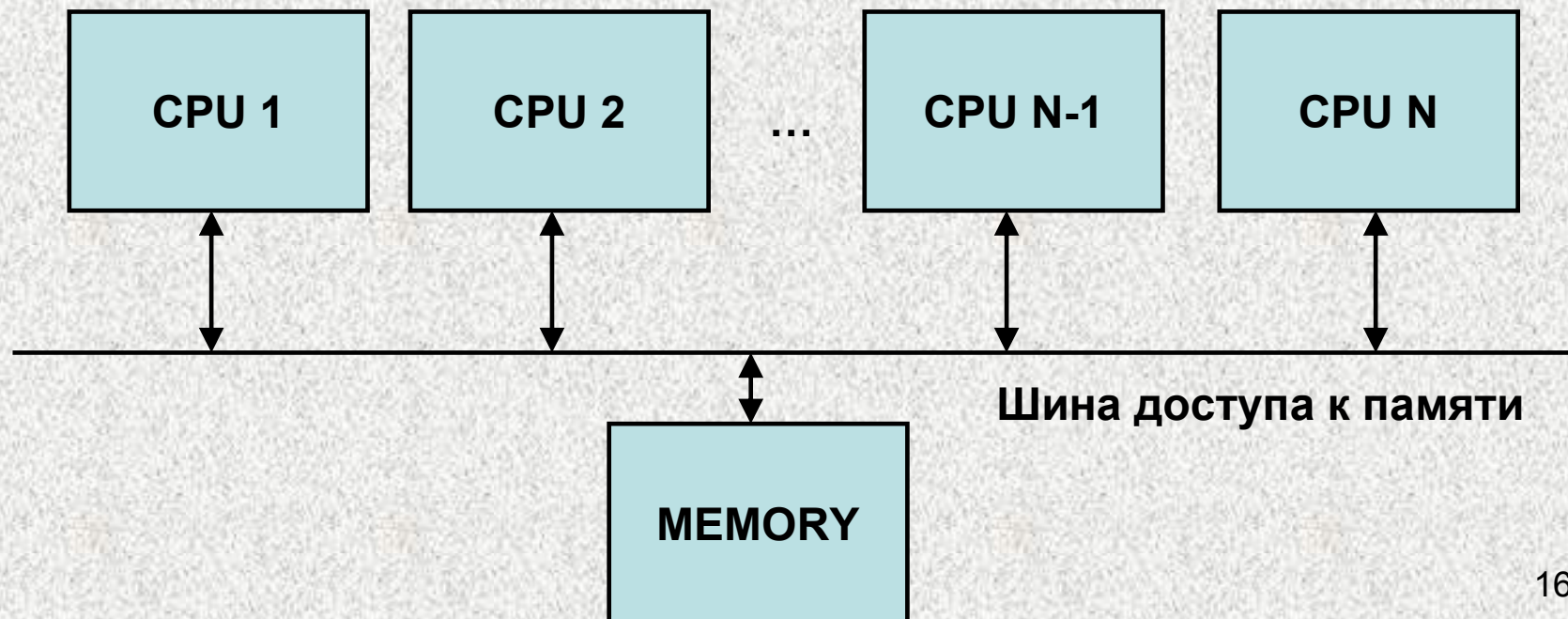


Многопроцессорный компьютер и многоядерный процессор

- С точки зрения программиста разница не видна.
- С точки зрения операционной системы разница проявляется в доступе процессоров и ядер к ресурсам (в частности к памяти – NUMA)
- Многоядерный процессор внутри себя решает те же проблемы, что и многопроцессорный компьютер (но возможно быстрее)

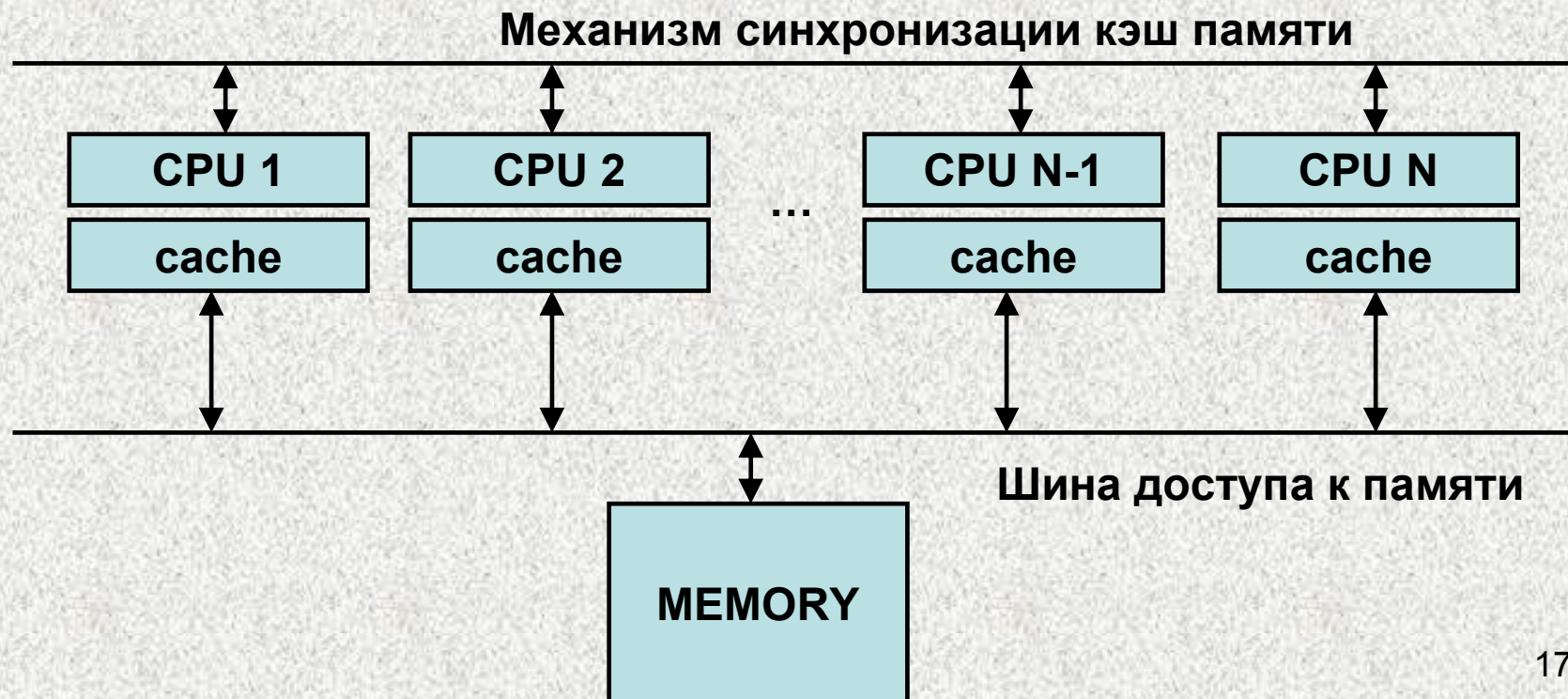
SMP - теоретически

- SMP пытается физически соответствовать логической памяти в программе
- любой процессор может выполнить любой процесс



SMP - практически

- Каждый процессор имеет свой кэш и страницы кэш периодически необходимо синхронизировать.



SMP - проблемы

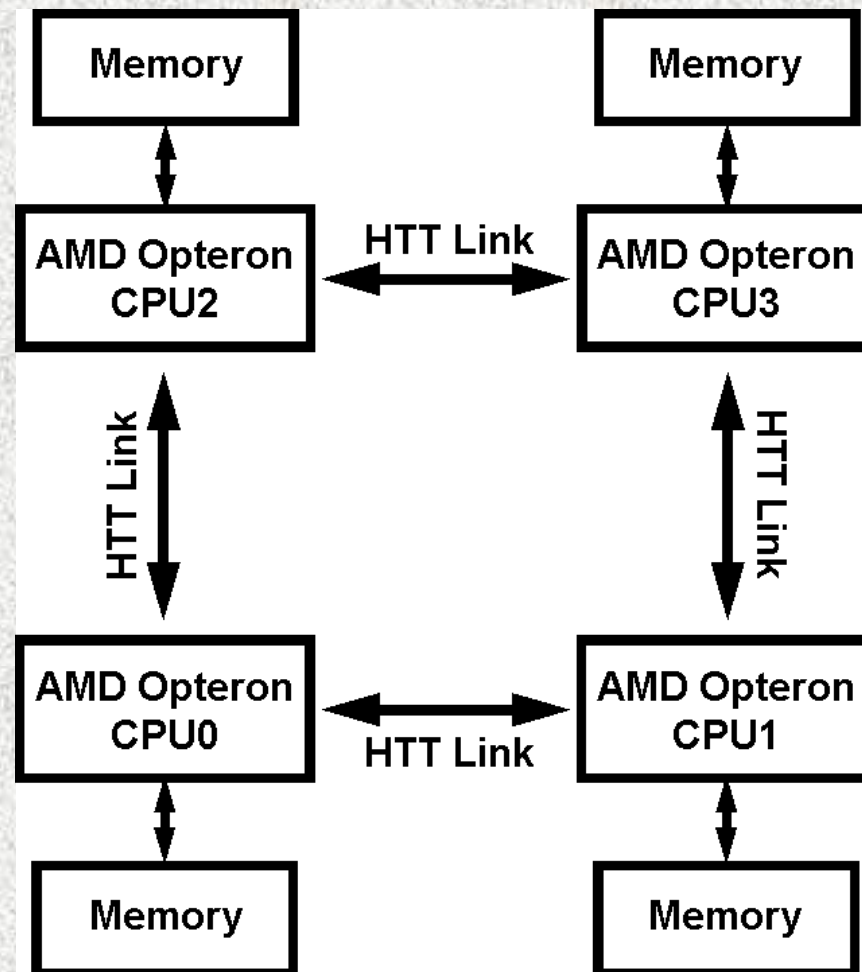
- С увеличением числа процессоров:
 - расходы на содержание кэш возрастают,
 - пропускная способность шины памяти делится на все процессоры,
 - переключение между задачами усложняется (доступ к общей памяти симметричный, а к кэш страницам нет)

NUMA

- **NUMA (Non-Uniform Memory Access)** схема реализации компьютерной памяти, используемая в мультипроцессорных системах, когда время доступа к памяти определяется её расположением по отношению к процессору.
- Не избавляет от необходимости заботиться о когерентности кэш'а. (**Cache coherent NUMA - ccNUMA**)

NUMA в исполнении AMD

- Каждый процессор имеет собственную память
- HyperTransport отвечает за синхронизацию кэш памяти



NUMA

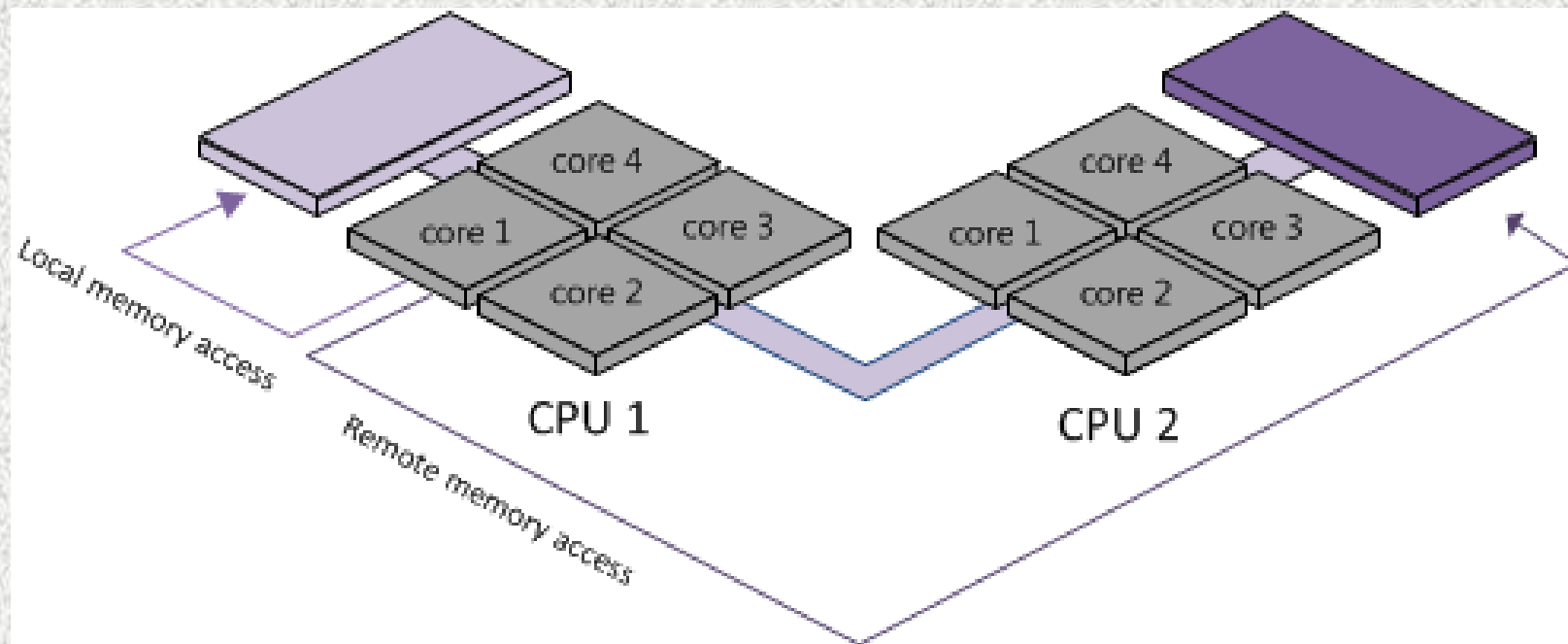
- Достоинства по сравнению с SMP:
 - У процессора есть «собственный» кусочек памяти, доступ к которому не делится с другими процессорами.
 - Появилась возможность связать процессор-процесс-данные в одно целое, поэтому при переключении задачи увеличивается шанс сохранить кэш памяти.
 - Общий объём памяти в компьютере легче увеличить благодаря распределению контроллеров памяти.

NUMA - проблемы

- С увеличением числа процессоров:
 - расходы на содержание кэш возрастают,
 - управление ресурсами и планирование ресурсов усложняется

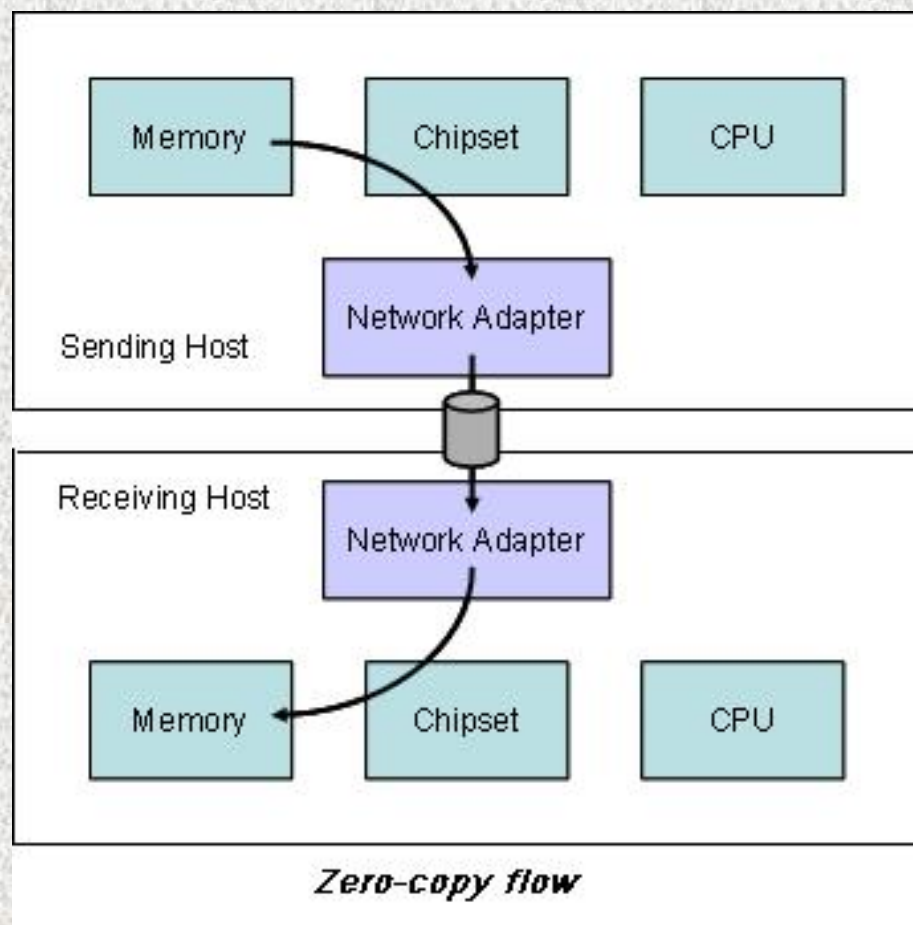
RDMA - remote direct memory access

- Идеализированное представление: virtual memory отображается чудесным образом в чужой компьютер.



RDMA - remote direct memory access

- На практике для реализации RDMA используются различные коммуникационные сети (вплоть до Ethernet).
- В идеальном случае RDMA сетевой адаптер работает асинхронно с процессором в системе.



RDMA - преимущества

- Нет необходимости заботиться о deadlock'ах
- Ассинхронные взаимодействия могут быть выгоднее синхронных (типичных для MPI приложений)
- Не требуется дополнительной памяти, все пересылки происходят от точки до точки.

RDMA недостатки

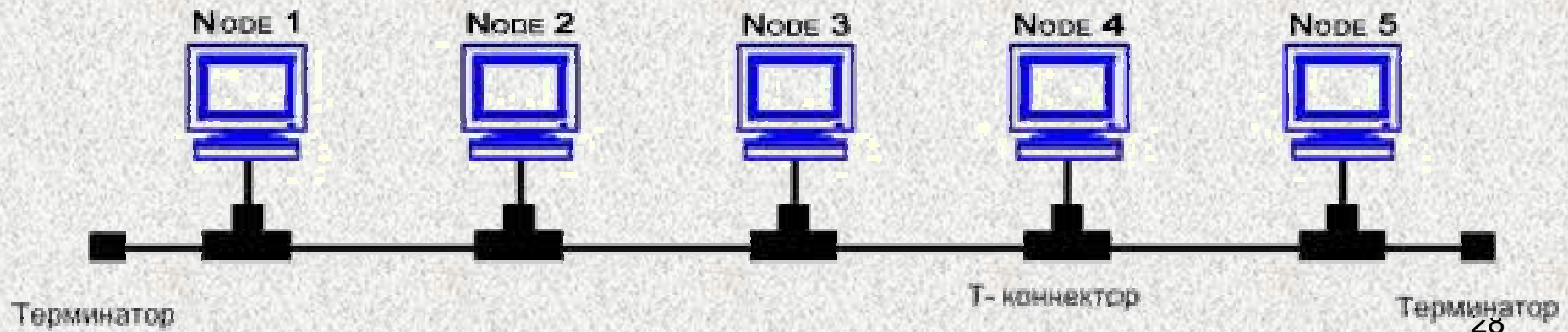
- RACE condition по прежнему остаётся и требует средств синхронизации процессов.
- К проблеме синхронизации кэш памяти (использующей очень быстрые каналы типа HyperTransport) добавляется проблема синхронизации rdma памяти (но на гораздо более медленных каналах)

Сетевые топологии

- Геометрическое расположение линий связи относительно узлов сети и физическое подключение узлов к сети называется физической топологией.
- Логическая топология определяет направления потоков данных между узлами сети и способы передачи данных.

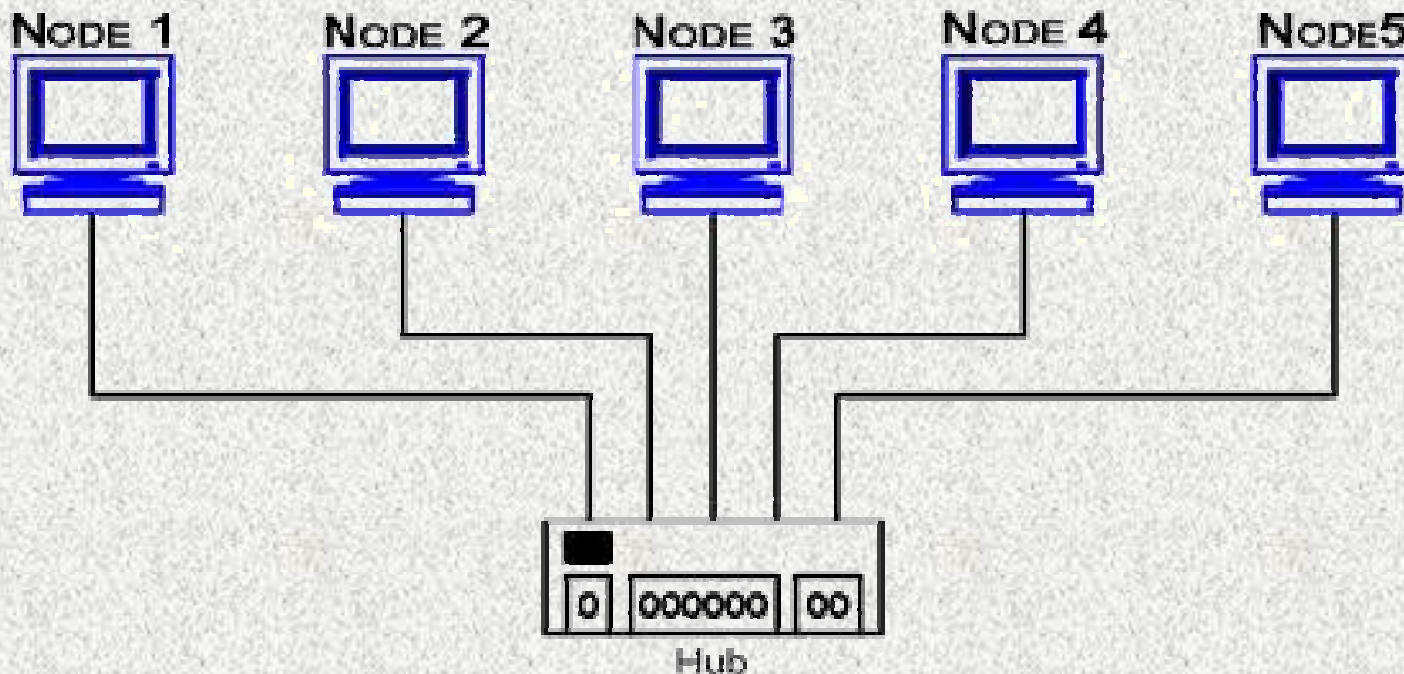
Шинная топология

- Такая физическая топология применяется в локальных сетях с архитектурой Ethernet (классы 10Base-5 и 10Base-2 для толстого и тонкого коаксиального кабеля соответственно).
- Такую логическую топологию имеют локальные сети Ethernet. Передача сигналов в такой топологии является широковещательной, т.е. сигналы распространяются одновременно во все направления .



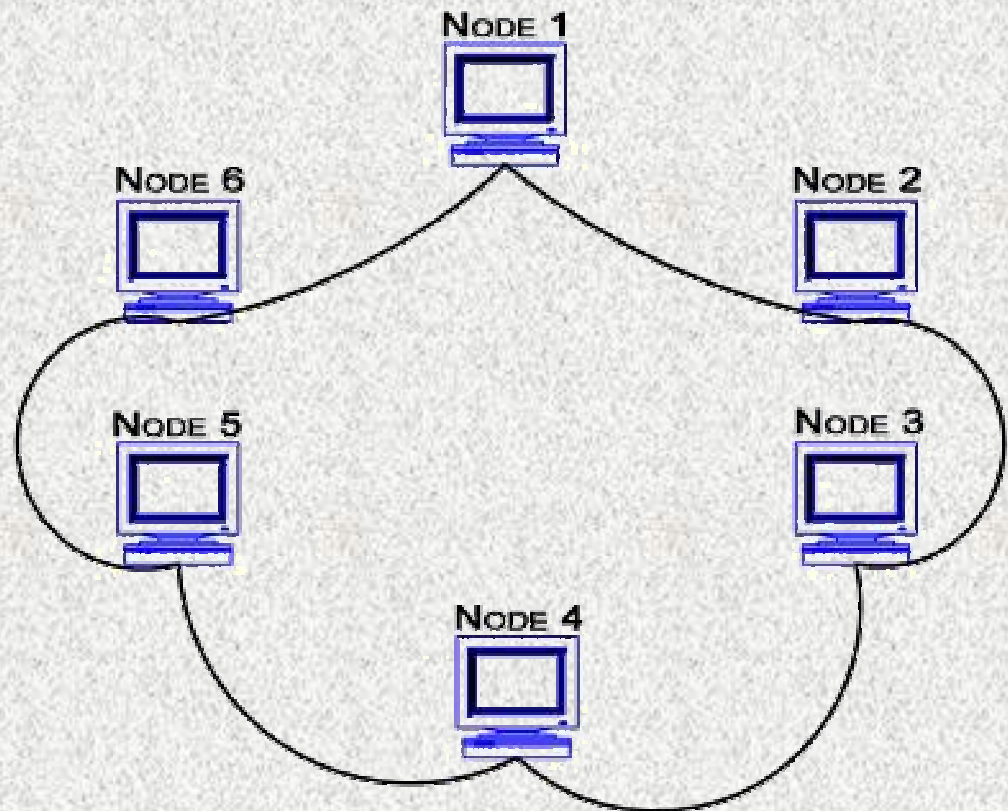
Топология типа «звезда»

- При физической топологии сети типа «звезда» каждый узел подсоединяется кабелем (витой парой) к концентратору или хабу (*hub*). (Пример Ethernet).
- Логическая топология типа «звезда» обеспечивает прямую передачу данных только между центральным узлом и остальными. (Пример NFS).



Топология “кольцо”

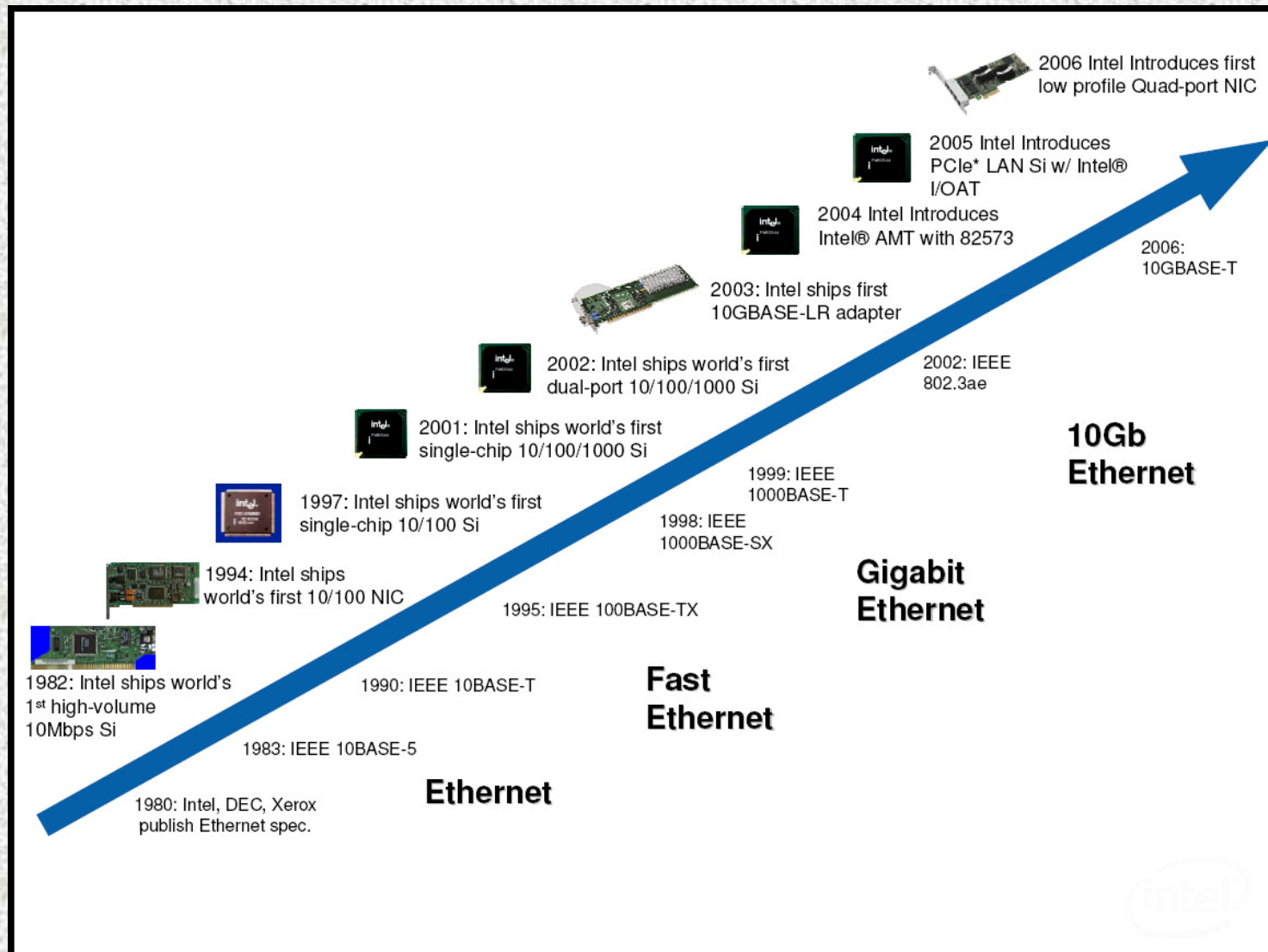
- В сети с физической топологией кольцо все узлы соединены каналами связи в неразрывное кольцо, по которому передаются данные. (Scalable Coherent Interface - SCI)



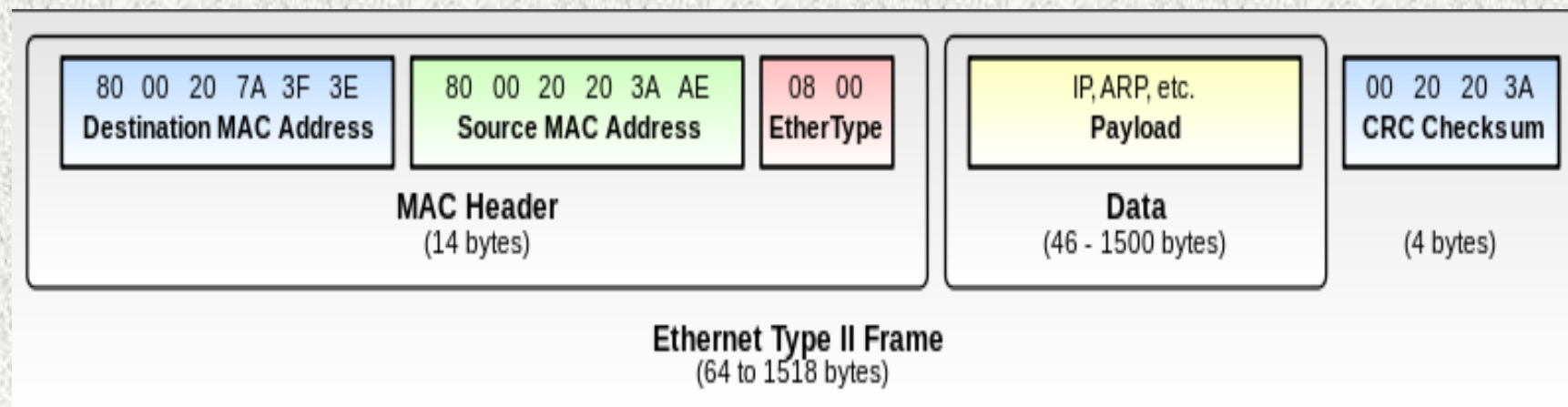
Ethernet

- Стандарт Ethernet предусматривает, что каждая сетевая карта (равно как и встроенный сетевой интерфейс) должна иметь уникальный шестибайтный номер (MAC-адрес), прошитый в ней при изготовлении. Этот номер используется для идентификации отправителя и получателя кадра.

Эволюция Ethernet



Ethernet кадр



Наиболее распространённый формат Ethernet кадра.

Payload содержит полезное содержимое кадра.

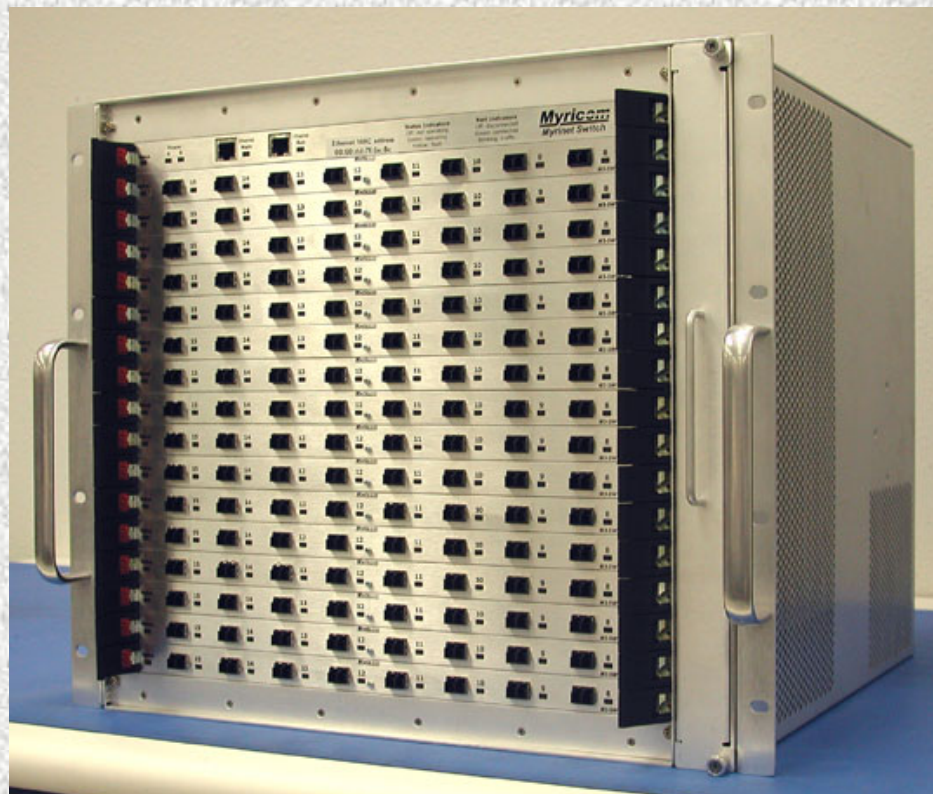
EtherType определяет тип полезного содержимого

Причины латентности Ethernet

- Случайная задержка перед посылкой кадра.
- Сохранение сообщения коммутатором
- Поиск по таблице MAC-адресов
- Повторная посылка сообщения (качественные коммутаторы начинают пересылку сообщения не дожидаясь конца приёма)
- Коллизии при одновременной посылке пакетов одной принимающей стороне
- Дополнительно от TCP/IP
 - Поиск адресата (преобразование IP в MAC)
 - Получение подтверждения о приёме сообщения

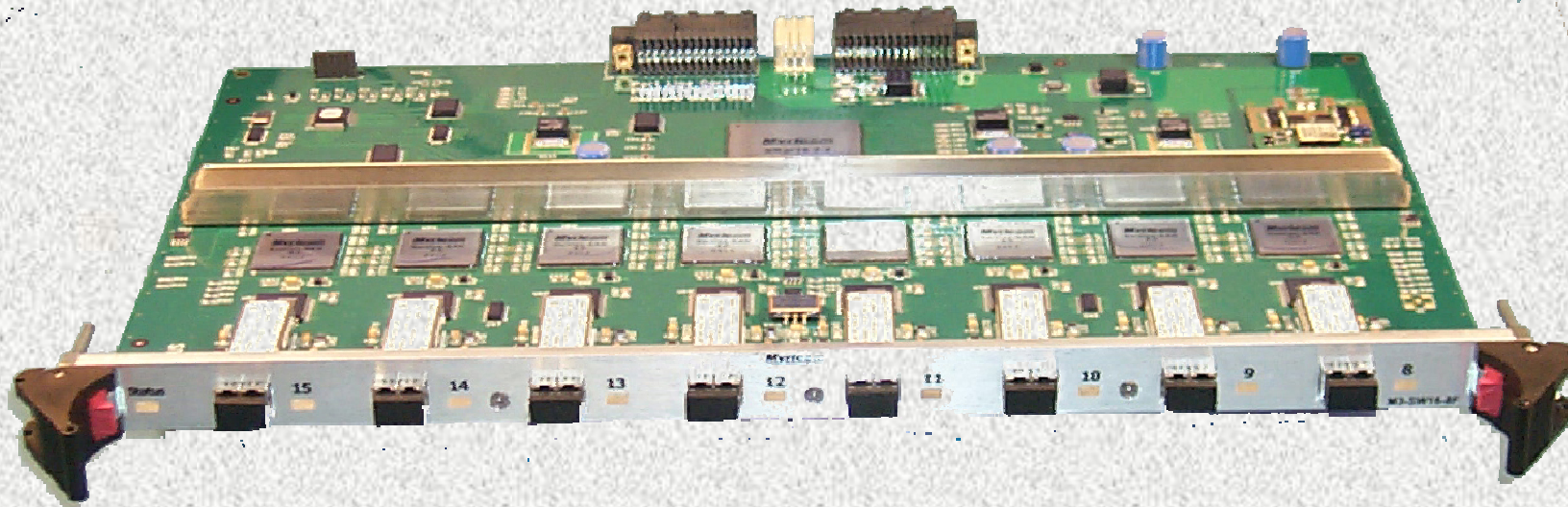
Myrinet коммутатор ВНЕШНИЙ ВИД

- Коммутаторы Myrinet строятся по модульной схеме. В основе системы 16ти или 8ми портовый кросс-бар



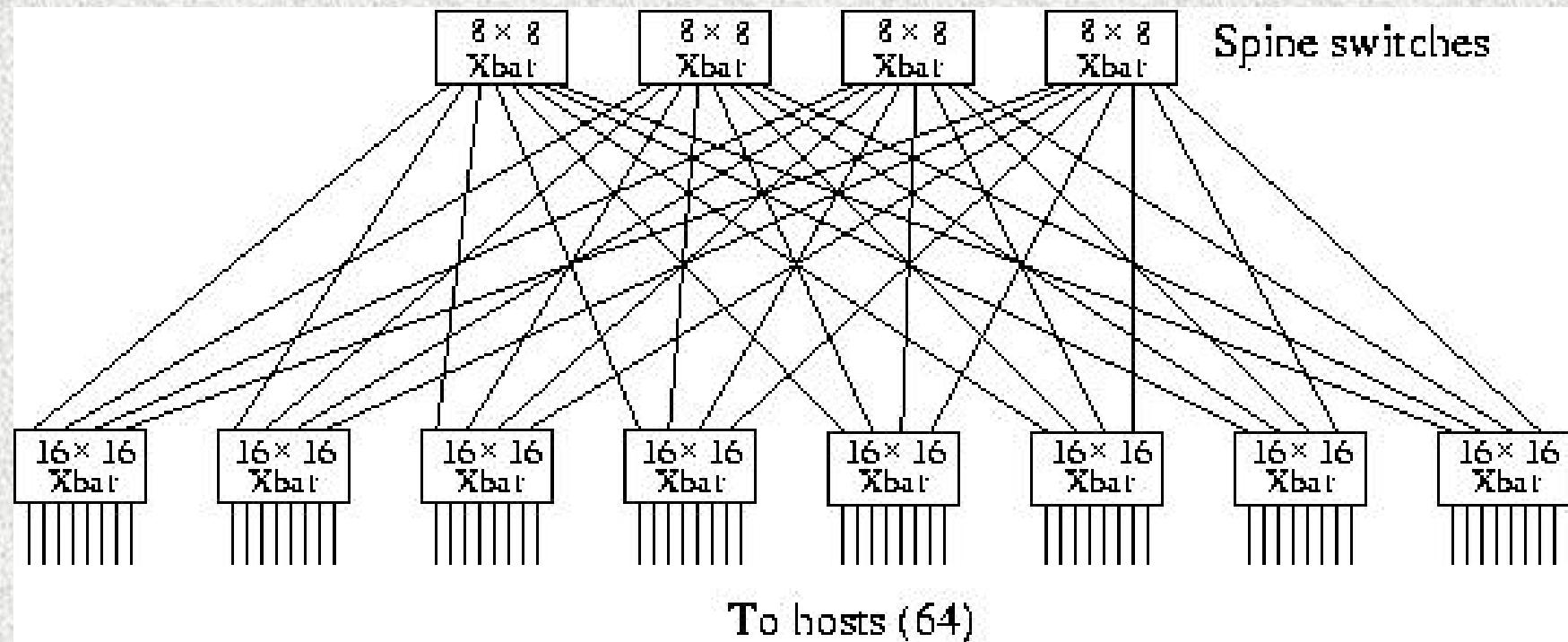
Line-карты Myrinet

- Существуют карты для: мониторинга, соединения коммутаторов, подключения клиентов



Кросс-бар

- Основа Myrinet – чип XBar, осуществляющий коммутацию по принципу телефонной станции



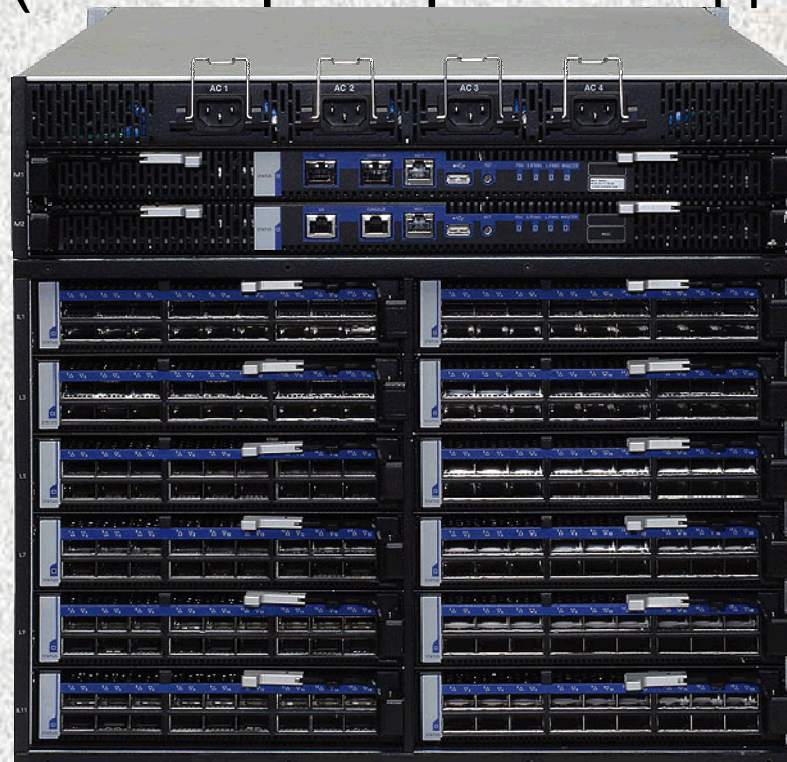
Кросс-бар Myrinet

- Для осуществления коммуникаций в Myrinet используется карта маршрутов.
- Каждый узел имеет свою копию (версию) карты, в которой прописываются «телефонные» адреса всех соседей.
- Предположение: Для одного коммутатора достаточно 3х цифрового маршрута: например маршрут «451» будет означать четвёртый Spine XBar, 1 порт на пятом конечном XBar'е.
- При передаче сообщения используется DMA из одной материнской платы в другую, без всяких промежуточных копирований данных!!!

Infiniband коммутатор ВНЕШНИЙ ВИД

- Коммутаторы Infiniband могут строиться по модульной схеме (но не обязательно). В основе системы (по непроверенным данным) 18ти и 36ти

портовый
кросс-бар



Infiniband - производительность

Пропускная способность интерфейса Infiniband, raw / data					
	SDR	DDR	QDR	FDR	EDR
1X	2,5 / 2 бит/с	5 / 4 Гбит/с	10 / 8 бит/с	14,0625 / 14 Гбит/с	25,78125 / 25 бит/с
4X	10 / 8 Гбит/с	20 / 16 Гбит/с	40 / 32 Гбит/с	56,25 / 56 Гбит/с	103,125 / 100 Гбит/с
12X	30 / 24 Гбит/с	60 / 48 Гбит/с	120 / 96 Гбит/с	168,75 / 168 Гбит/с	309,375 / 300 Гбит/с

Infiniband Messages

- InfiniBand transmits data in packets of up to 4 KB that are taken together to form a message. A message can be:
 - a direct memory access read from or, write to, a remote node (RDMA)
 - a channel send or receive
 - a transaction-based operation (that can be reversed)
 - a multicast transmission.
 - an atomic operation

Infiniband отличие от Myrinet

- Поддержка RDMA
- Поддержка multicast transmission.
- Быстродействие
- Больше количество производителей
- Открытый код основного программного обеспечения OFED

Взаимодействие на общей памяти vs. передача сообщений

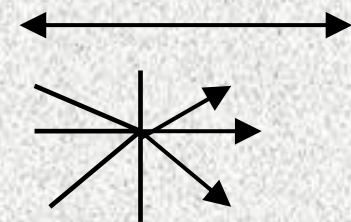
- Общая память, основные проблемы:
 - RACE CONDITION
 - Cache coherence
- Основные проблемы при передаче сообщений:
 - DeadLocks (взаимная блокировка процессов)
 - Затраты на синхронизацию процессов

Топология выч. процессов

- Начало программы
- Конец программы
- Единичный последовательный процесс
- Посылка/прием сообщений
- Барьер/групповые операции

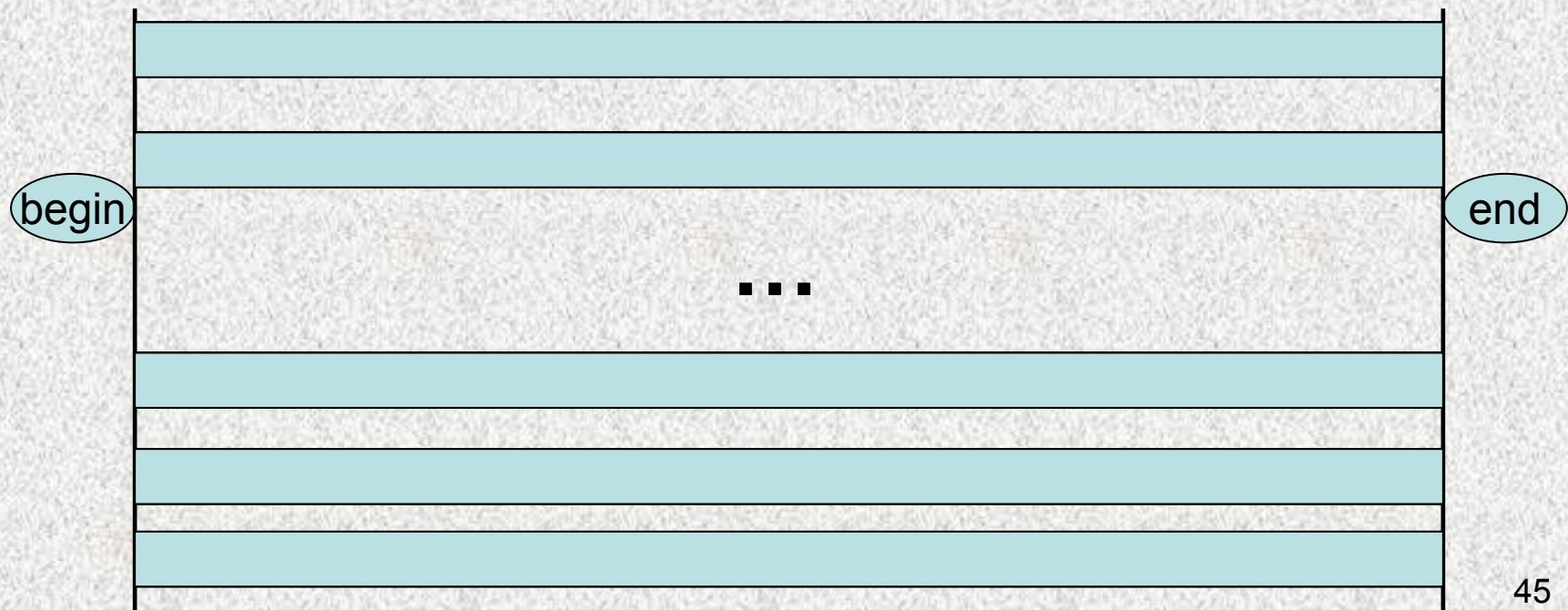
begin

end



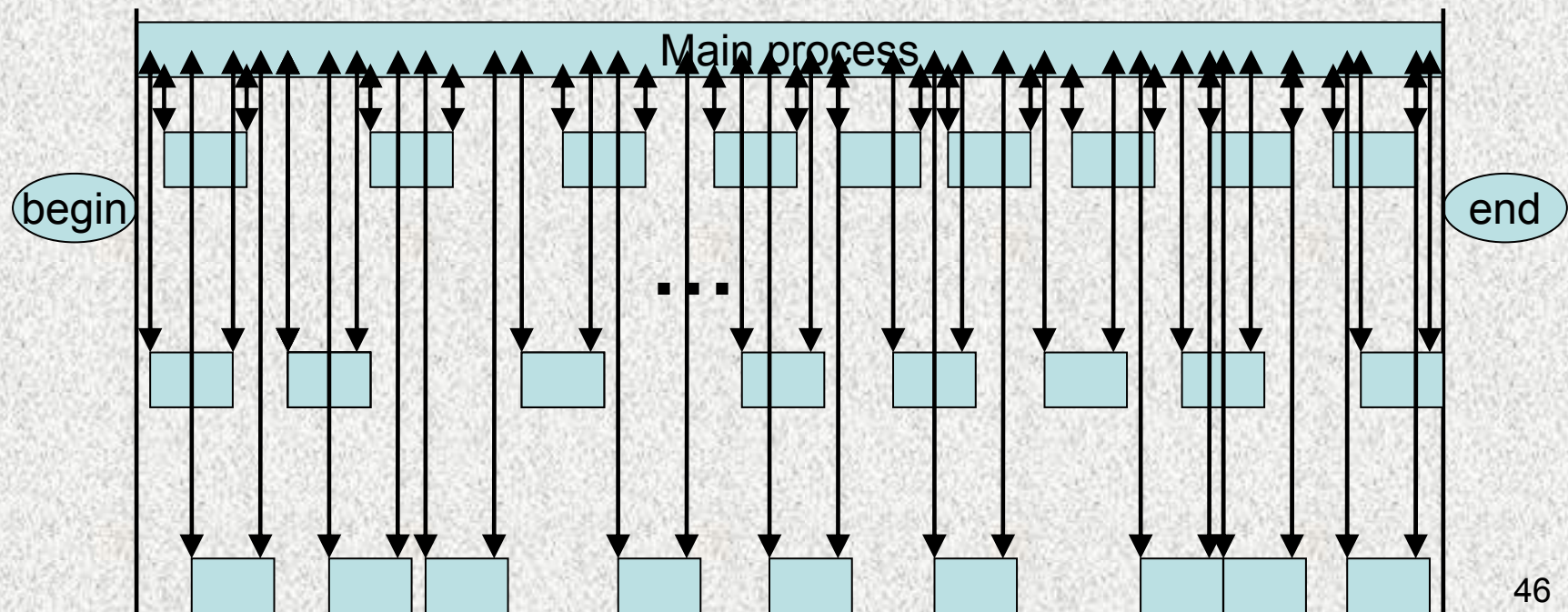
Pure parallel program

- Идеальный вариант параллельной программы. В жизни встречается, в случае, когда существует параллелизм по данным (например поиск фотографии в базе данных по фрагменту)



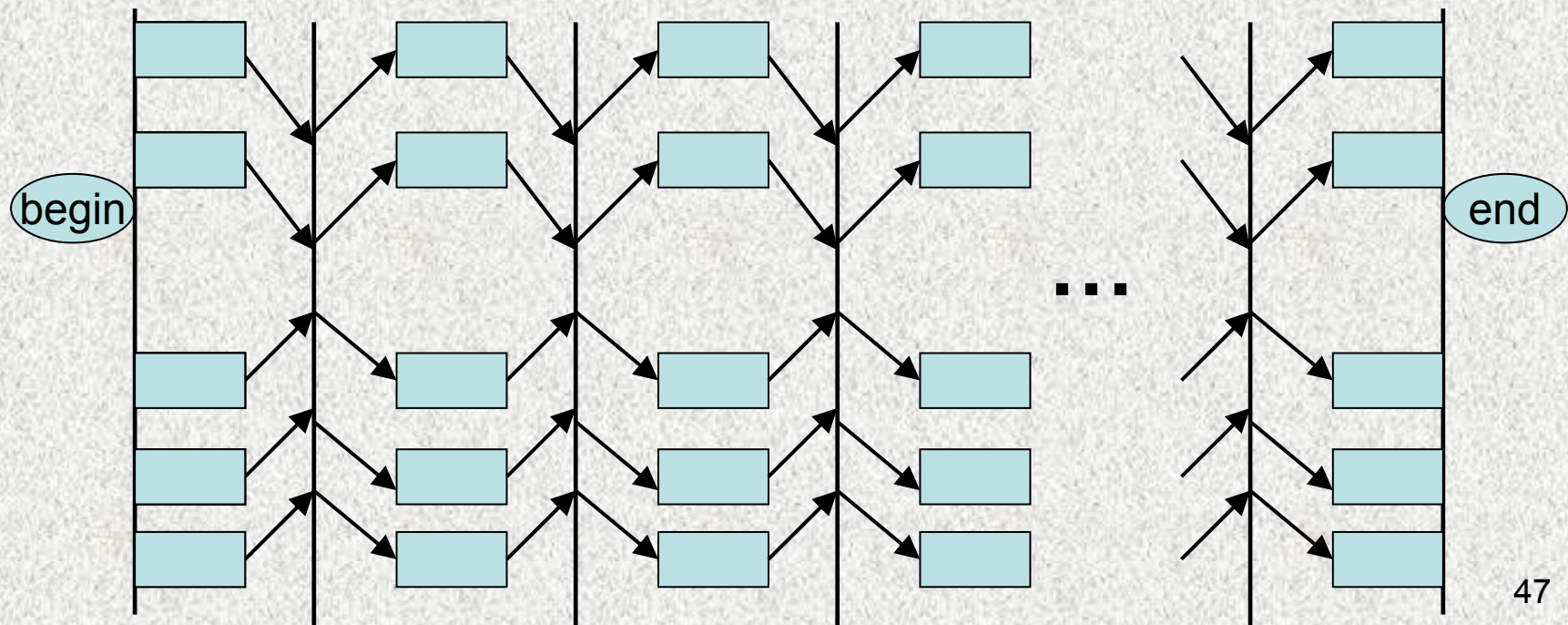
Client – server template

- Подходит для обработки массивов накопленных данных, главный процесс нужен для распределения задач между остальными узлами.



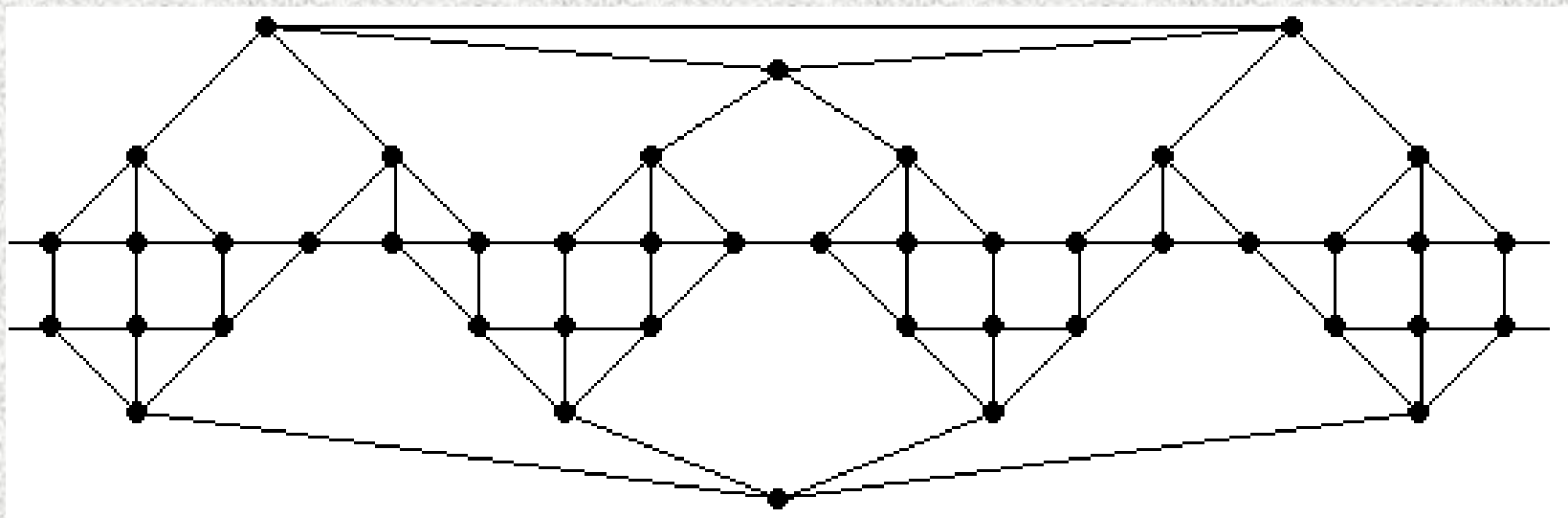
Fixed topology template

- «Обыкновенный» итерационный процесс расчёта на сетке.
- К примеру можно использовать **MPI_Allgather**



Сложная топология вычислений

- Обычно сложная топология вычислительного процесса определяется данными. (Вычисления длительные, значит в топологии длительное время выполняются связующие отношения.)
- Приведите пример, когда сложная топология не связана с данными?



Коммуникатор в MPI

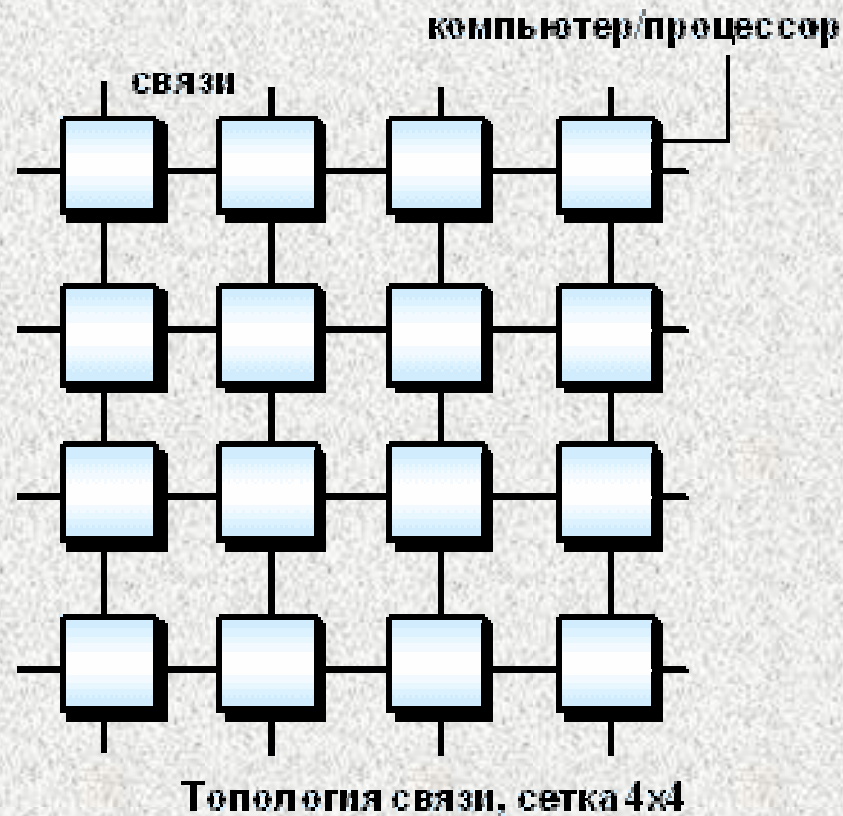
- Коммуникатор в MPI - специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров (контекст), используемых при выполнении операций передачи данных.
- Основным коммуникатором для всех MPI версий является коммуникатор MPI_COMM_WORLD, он предопределен извне программы.
- Для создания нового коммуникатора MPI необходим исходный коммуникатор.
- Варианты новых коммуникаторов:
 - Пользовательская группировка процессов
 - Виртуальная топология
 - Интеркоммуникатор

Топологии в MPI

- Топология - это механизм сопоставления процессам некоторого коммутатора альтернативной схемы адресации. В MPI топологии виртуальны, то есть они не связаны с физической топологией коммуникационной сети. Топология используется программистом для более удобного обозначения процессов, и таким образом, приближения параллельной программы к структуре математического алгоритма. Кроме того, топология может использоваться системой для оптимизации распределения процессов по физическим процессорам используемого параллельного компьютера при помощи изменения порядка нумерации процессов внутри коммутатора.
- Существуют два основных типа виртуальной топологии, которая может быть создана MPI, - топология сетки, и топология графа. Первый тип является подмножеством второго. Однако из-за активного использования сеток в приложениях выделяется отдельное подмножество функций MPI, цель которого - манипуляция виртуальными сетками.

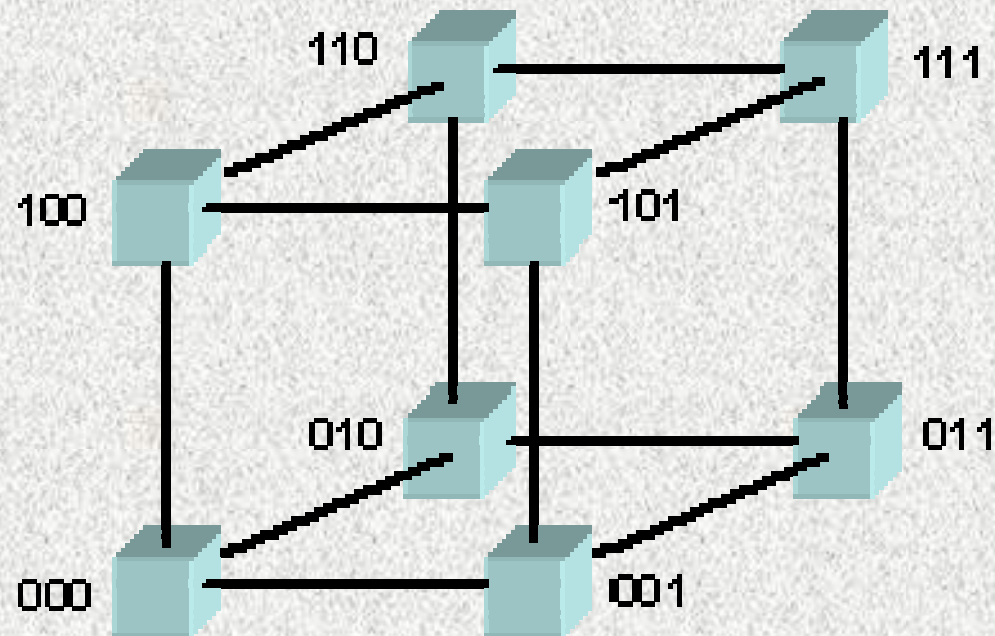
Топология в MPI

- Пример сетки. В двух измерениях без зацикливания



Топология в MPI

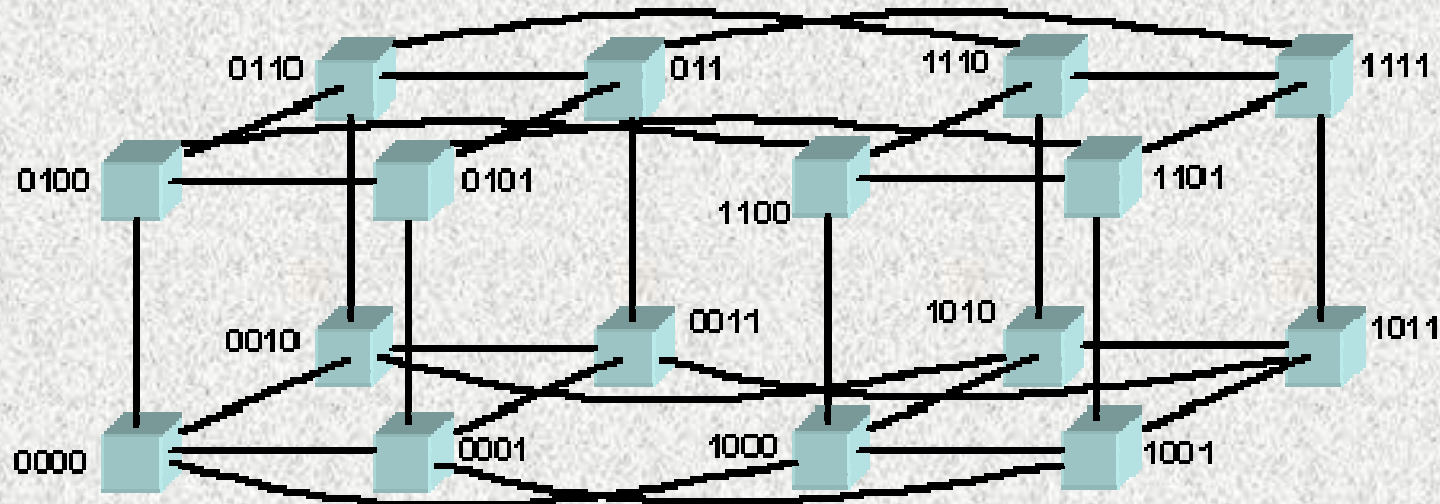
- Пример сетки. В трёх измерениях без зацикливания



Топология связи, 3-х мерный гиперкуб

Топология в MPI

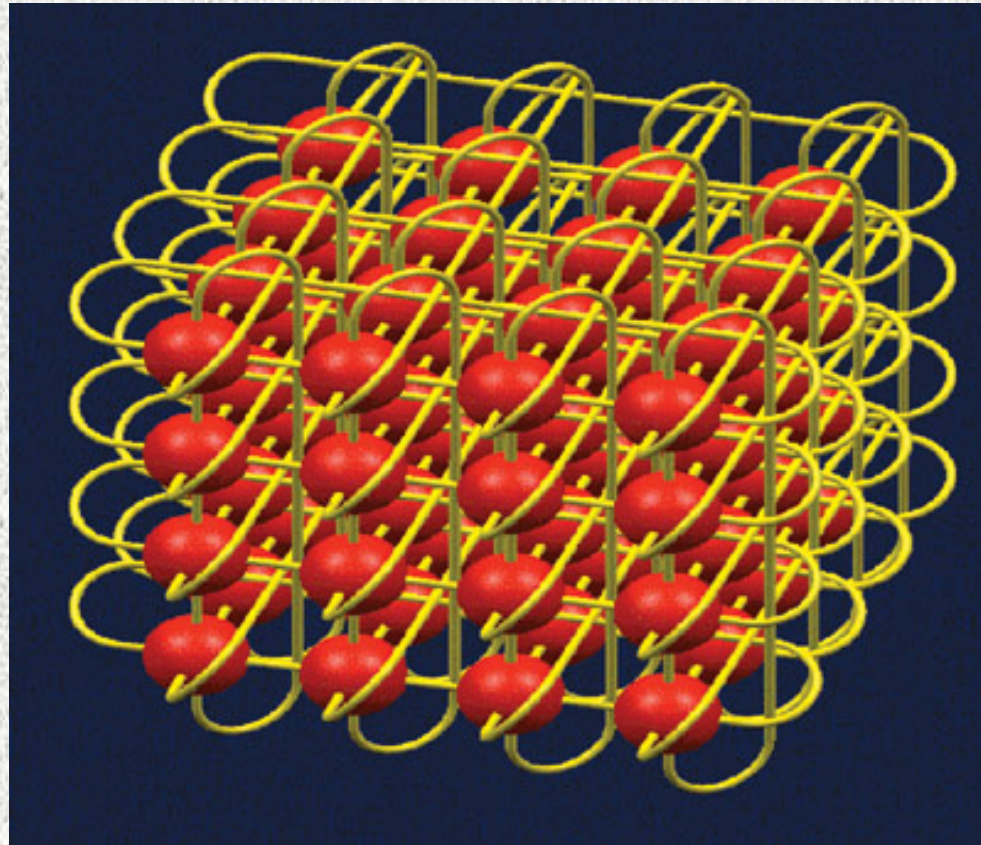
- Пример сетки. В четырёх измерениях без зацикливания



Топология связи, 4-х мерный гиперкуб

Топология в MPI

- Пример сетки. В трёх измерениях с зацикливанием



Сложный вычислительный процесс (MPI + SMP или NUMA)

- Недостаток:
 - Сложность системы растёт => Количество возможных ошибочных ситуаций увеличивается.
- Преимущество:
 - Можно использовать параллельные библиотеки, например BLAS.
 - ???

THE END