

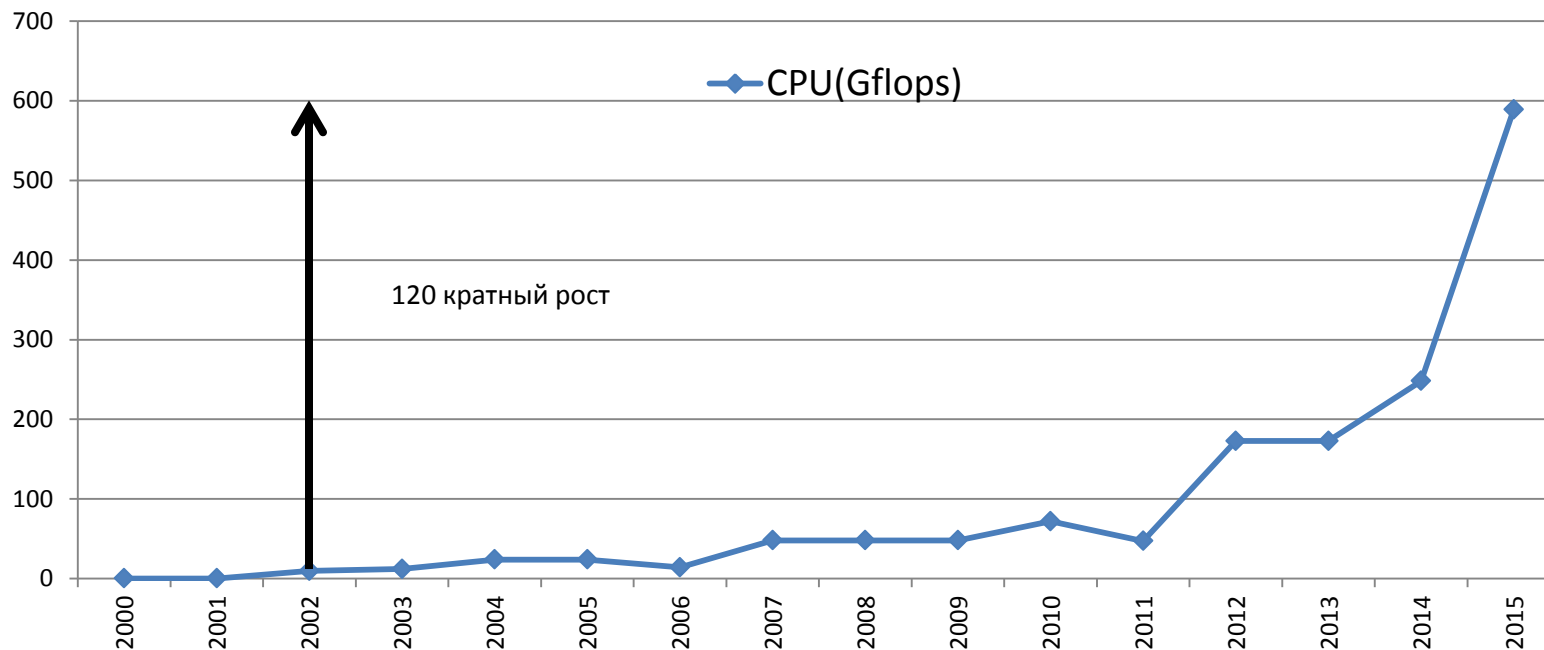
*Параллельные вычисления
(часть 1)*

Якобовский Михаил Владимирович

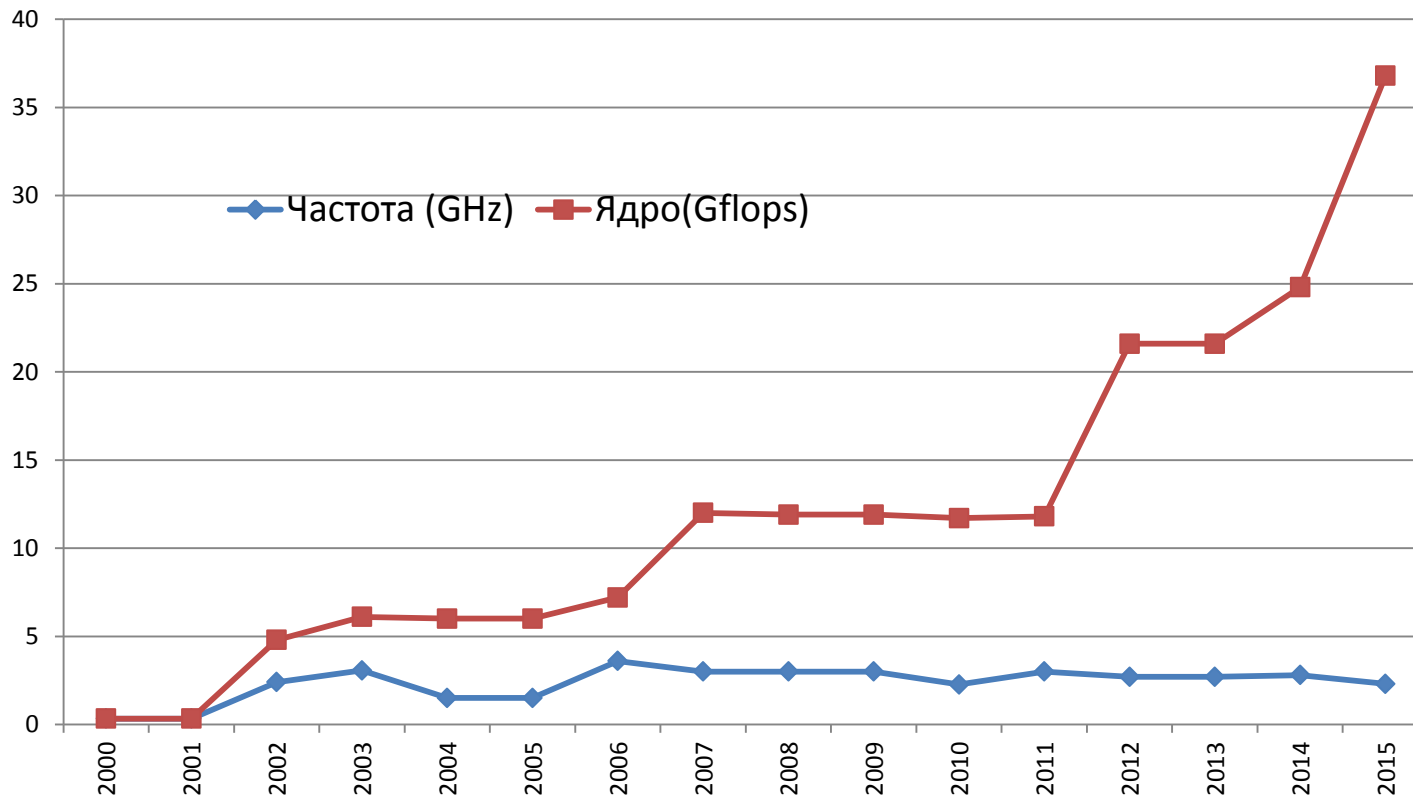
Содержание лекции

- ❑ Многопроцессорные системы
 - с распределенной памятью
 - с общей памятью
 - Гибридные
- ❑ Модели выполнения программ
- ❑ Свойства канала передачи данных
- ❑ Методы взаимодействия процессов
 - Методы передачи данных
 - Семафоры
- ❑ Ускорение и эффективность параллельных алгоритмов
- ❑ Пример алгоритма с низкой эффективностью, но высоким быстродействием

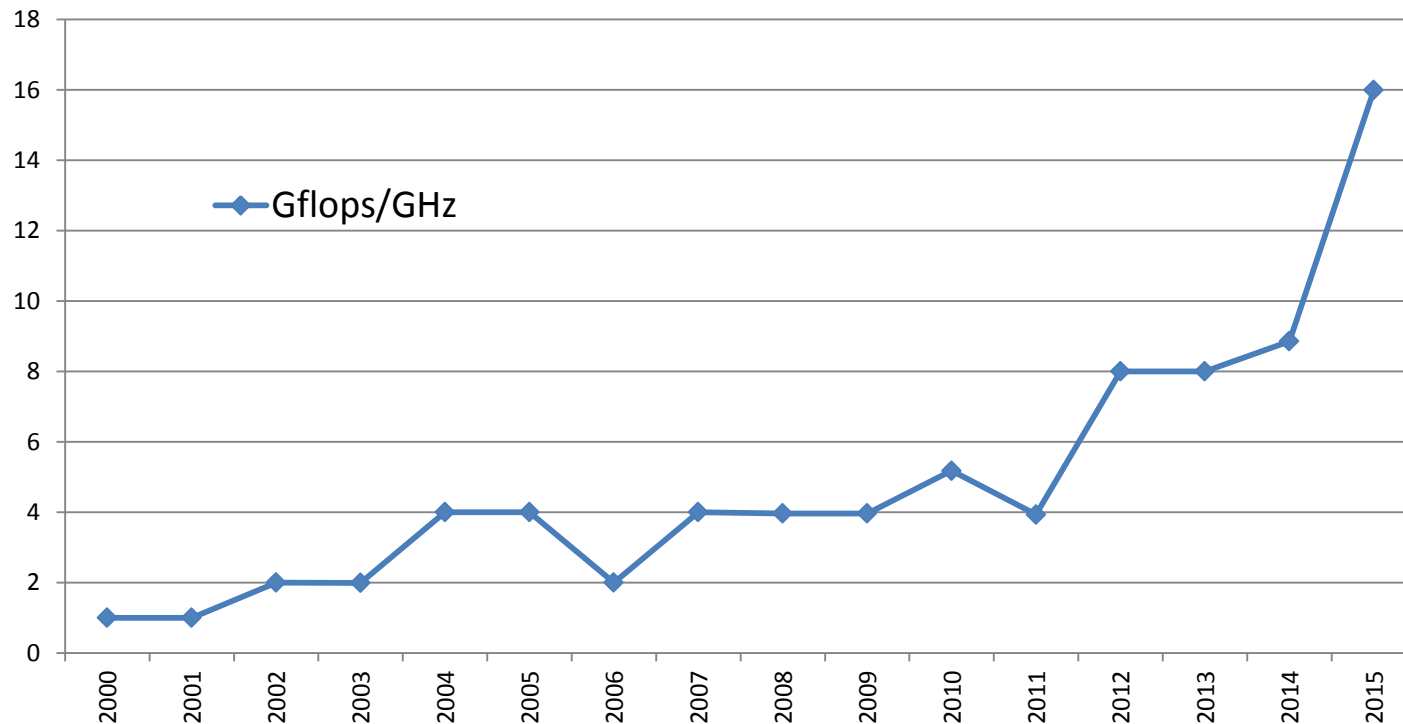
Производительность процессора



Производительность и тактовая частота процессорного ядра



Число операций, выполняемых процессорным ядром за один такт



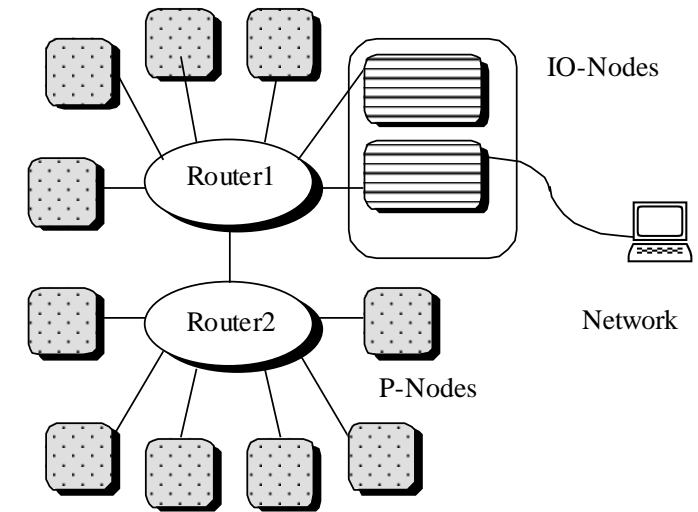
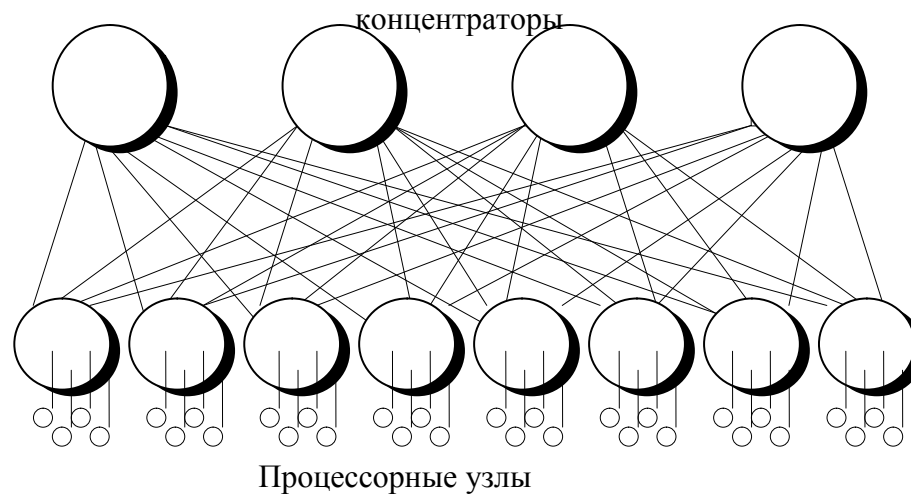
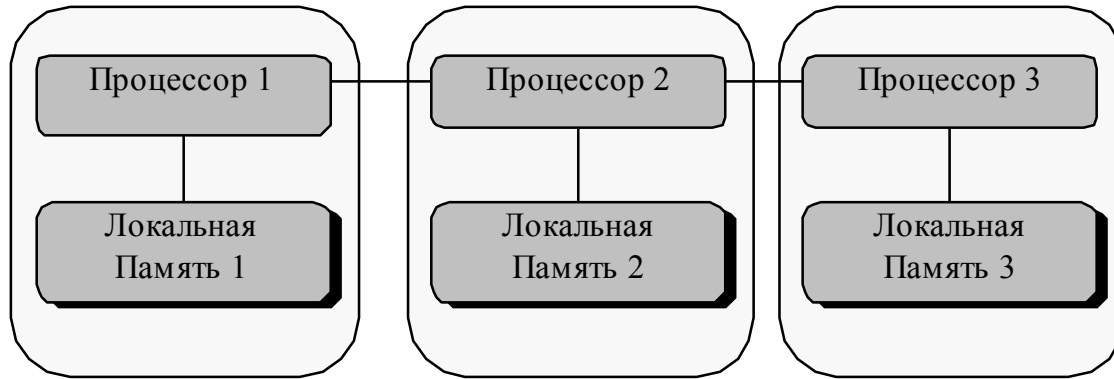
Некоторые системы top500.org

Суперкомпьютер	Cores	Rmax Tflops	Rpeak Tflops	Power MW	Процессор
Sunway TaihuLight	10 649 600	93 014	125 436	15.37	Sunway SW26010 260C 1.45GHz
K computer	705 024	10 510	12 280	12.66	SPARC64 VIIIfx 8C 2GHz
Tianhe-2 (MilkyWay-2)	3 120 000	33 862	54 902	17.81	Intel Xeon E5-2692v2 12C 2.2GHz Intel Xeon Phi 31S1P 57C 1.1GHz
Sequoia, BlueGene/Q	1 572 864	16 325	20 132	7.89	Power BQC 16C 1.60 GHz
Lomonosov 2	42 688	2 102	2 962	1.08	Xeon E5-2697v3 14C 2.6GHz

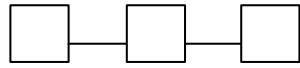
Время мощных вычислительных систем, имеющих в своем составе только один процессор прошло

- Вычислительная мощность суперкомпьютеров огромна, но:
 - ***Она образована суммой мощностей множества исполнительных устройств***
- На протяжении многих лет увеличение производительность компьютера автоматически означало снижение времени работы существующих программ. Теперь это не так:
 - ***Последовательные программы не могут работать на суперкомпьютерах быстрее, чем на однопроцессорных (одноядерных) компьютерах***

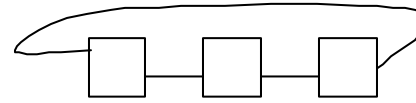
Многопроцессорные системы с распределенной памятью



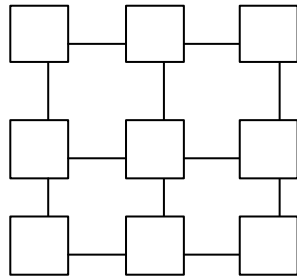
Топологии вычислительных систем



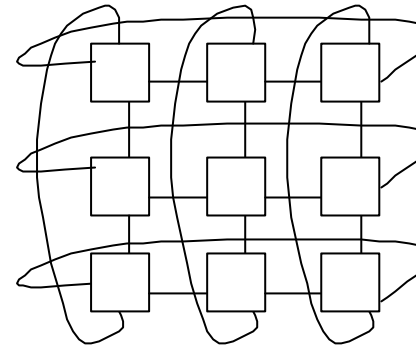
Топология «линейка»



Топология «кольцо»



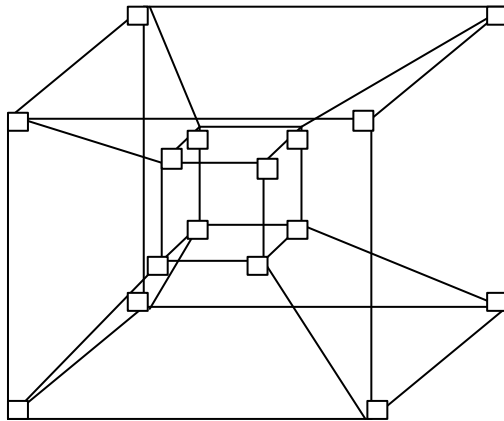
Топология «решетка 3x3»



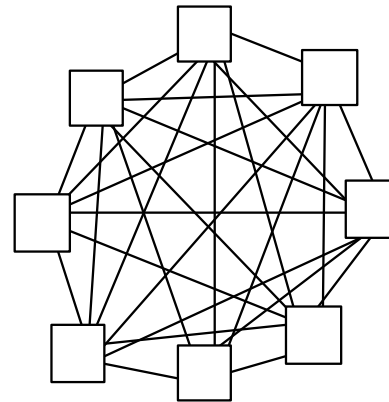
Топология «тор 3x3»

- Диаметр графа – максимальная длина кратчайшего пути в графе
- Центр графа – такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных
- Радиус графа – длина максимального пути от его центра

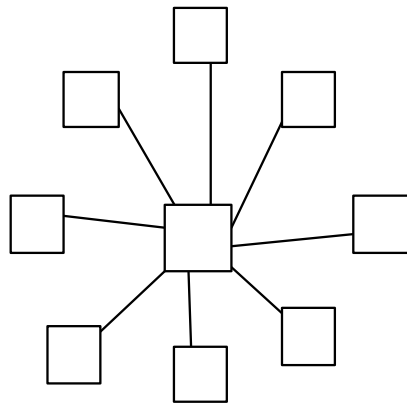
Топологии вычислительных систем



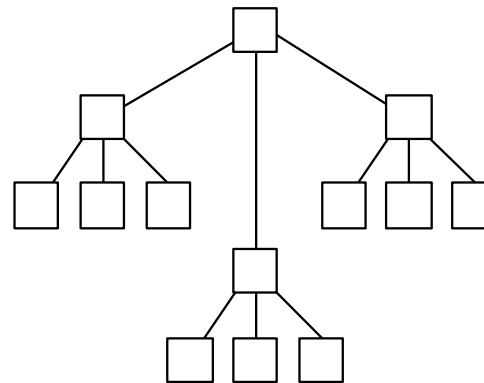
Топология «гиперкуб степени 4»



Топология «клика»

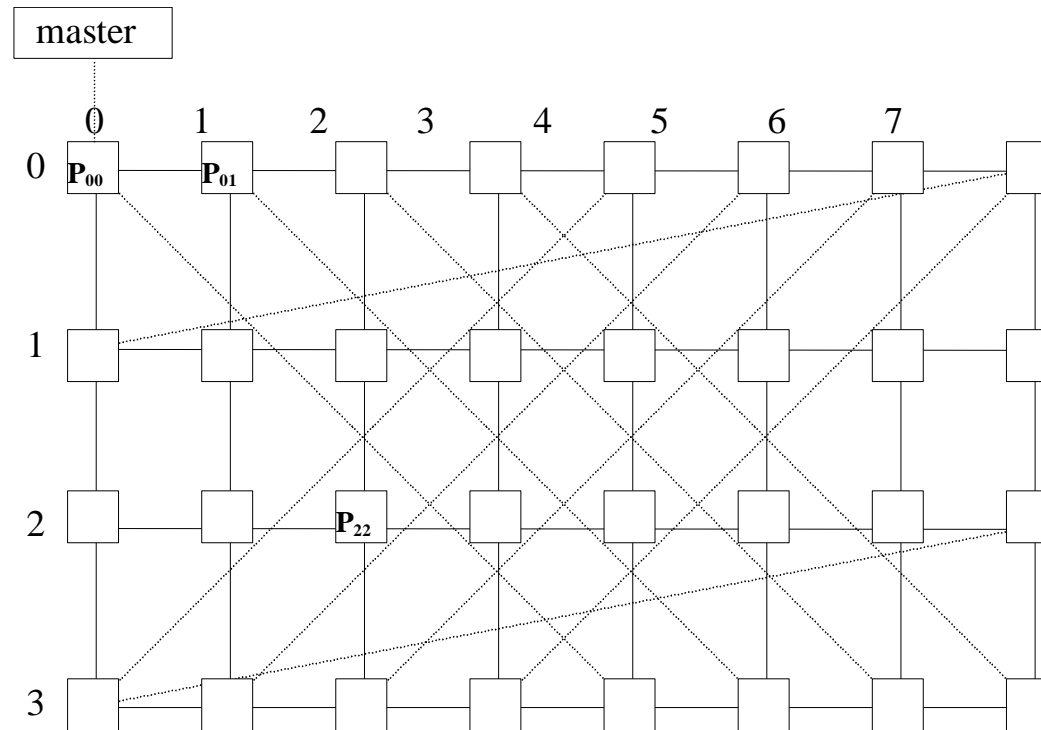


Топология «звезда»



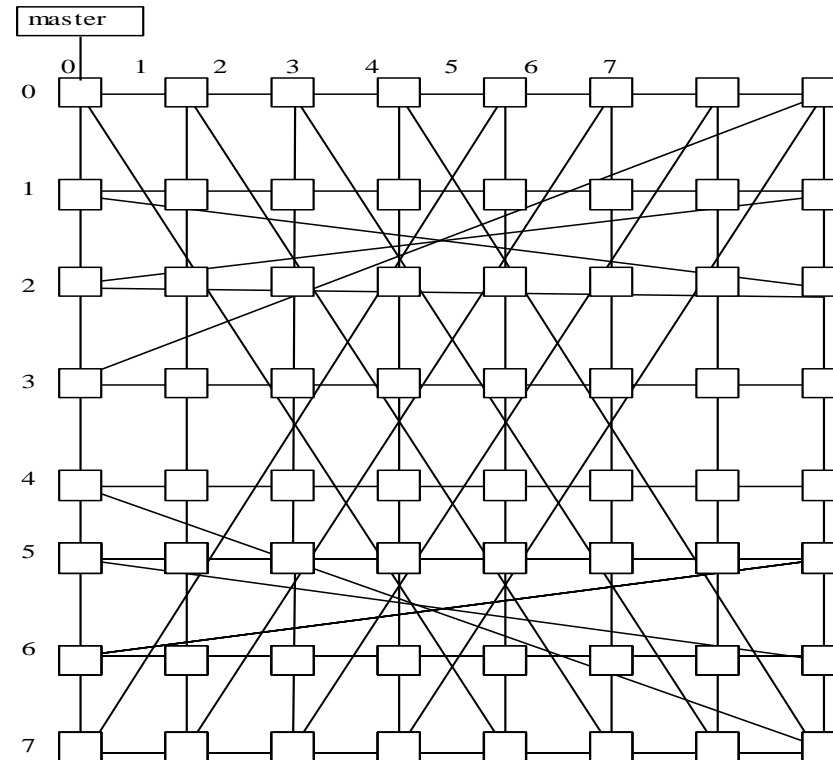
Топология «троичное дерево»

Минимизация радиуса графа



Пример графа из **32** процессоров с диаметром и радиусом равными **4**

Минимизация радиуса графа



Пример графа “пирамида” из **64** процессоров с диаметром и радиусом равными **6**

Многомерный тор

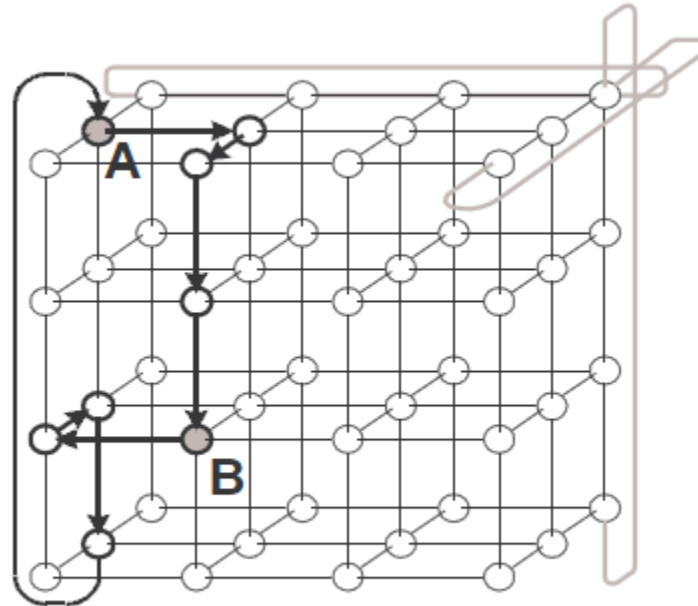
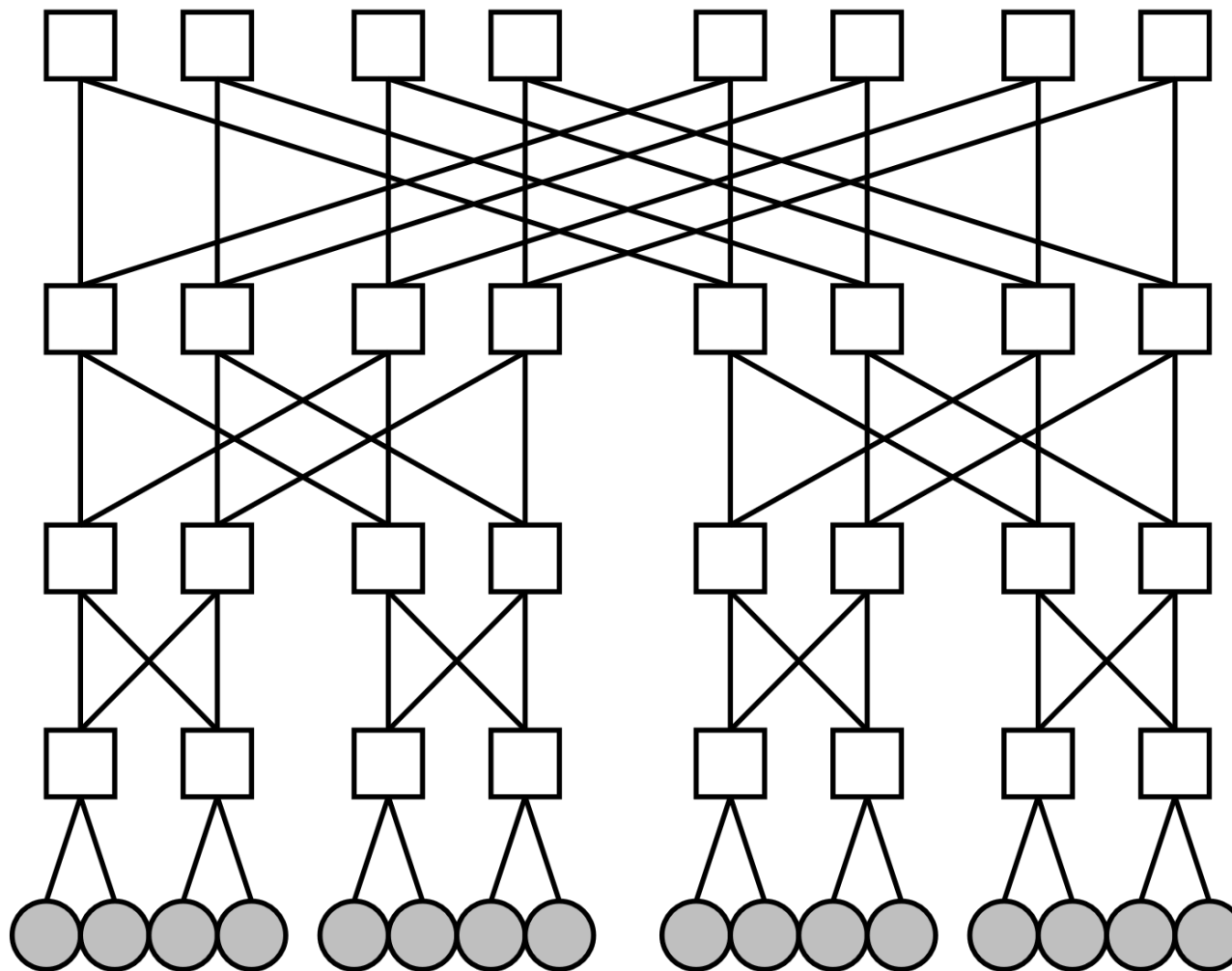


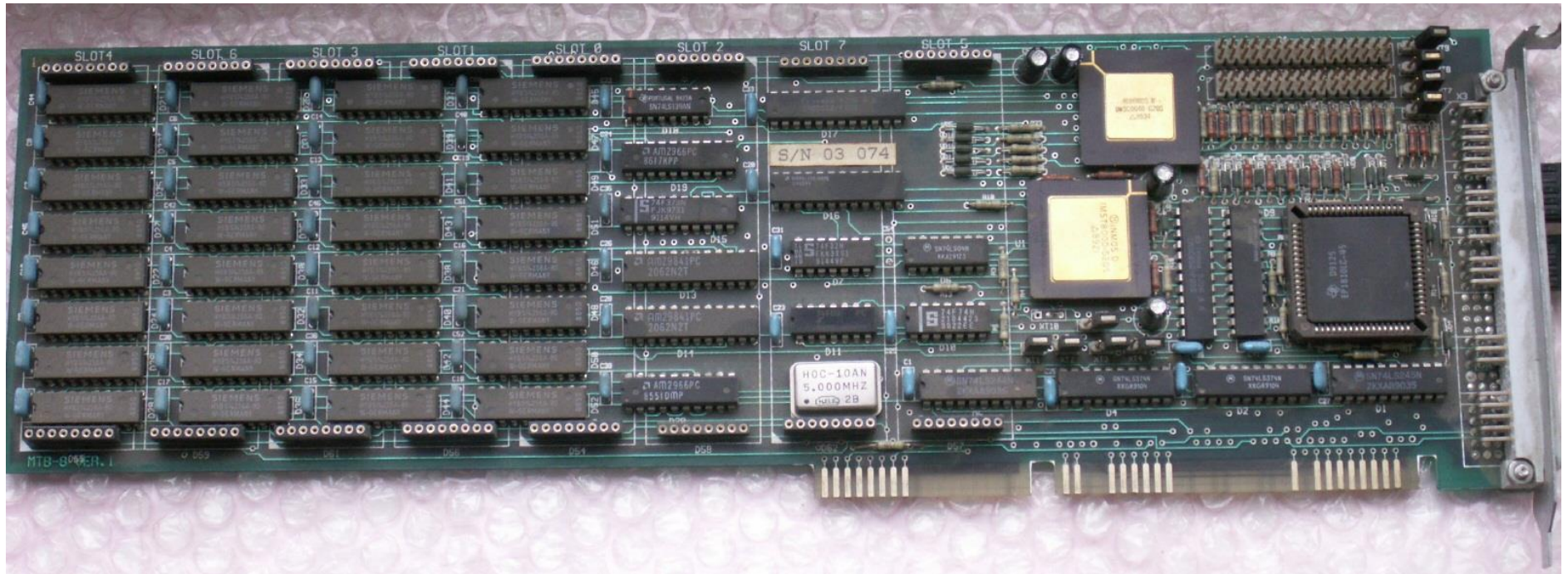
Рис. 3.4. Соединение узлов в трехмерный тор в сети SCI

Воеводин Вл.В., Жуматий С.А.
"Вычислительное дело и кластерные системы".
М.: Изд-во МГУ, 2007. - 150 с. ISBN 978-5-211-05440-0.

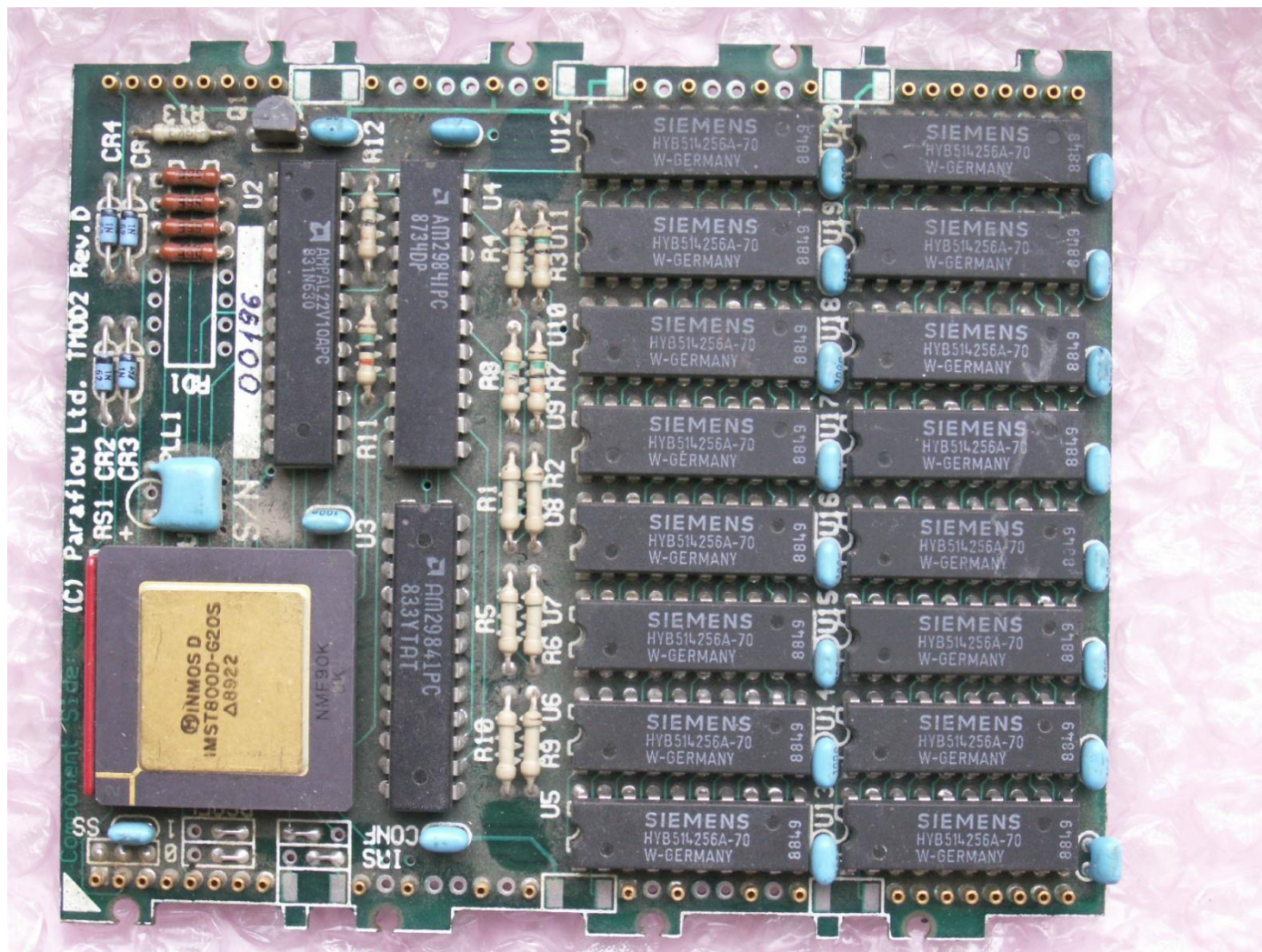
Толстое дерево



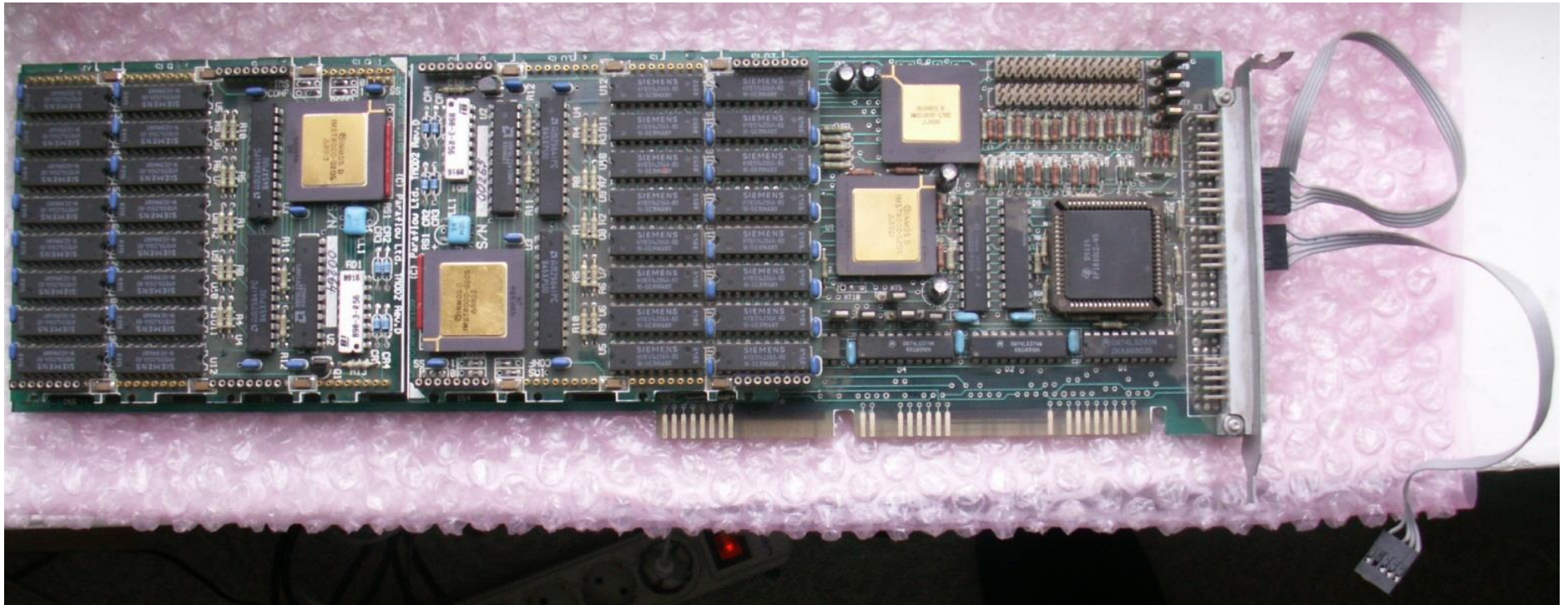
Транспьютерная материнская плата МТБ-8



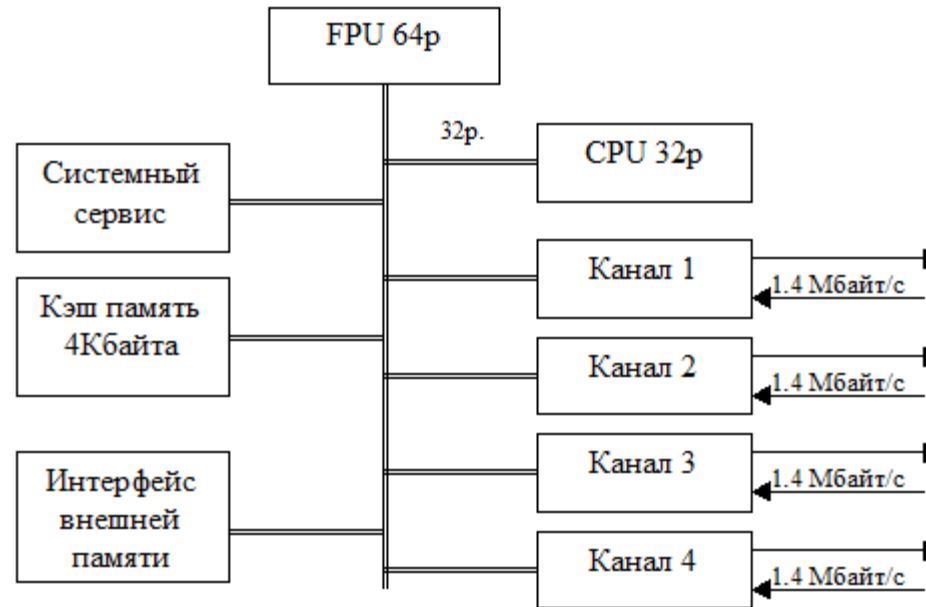
Транспьютер и оперативная память



Три транспьютера на плате МТБ-8

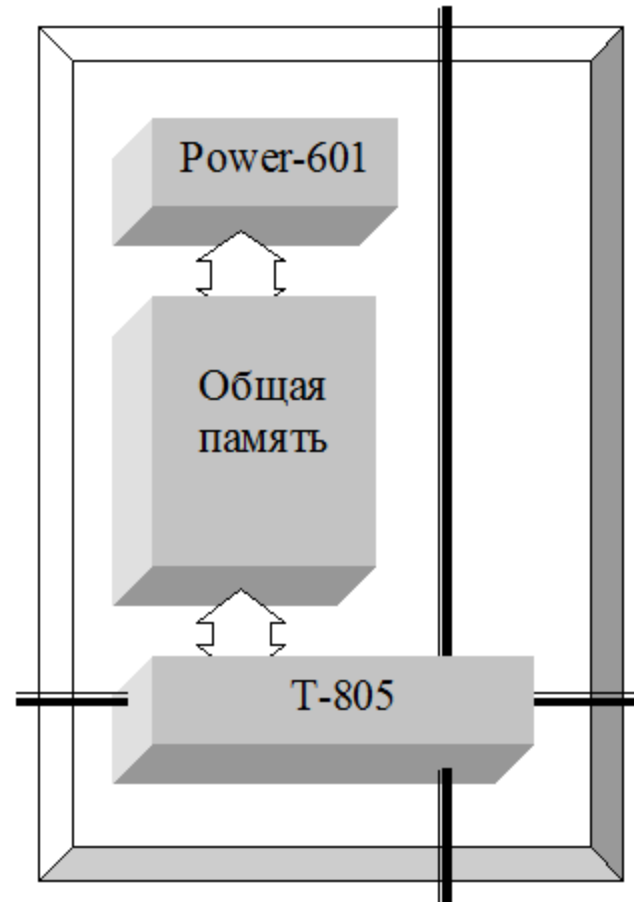


Структура транспьютера Т-800



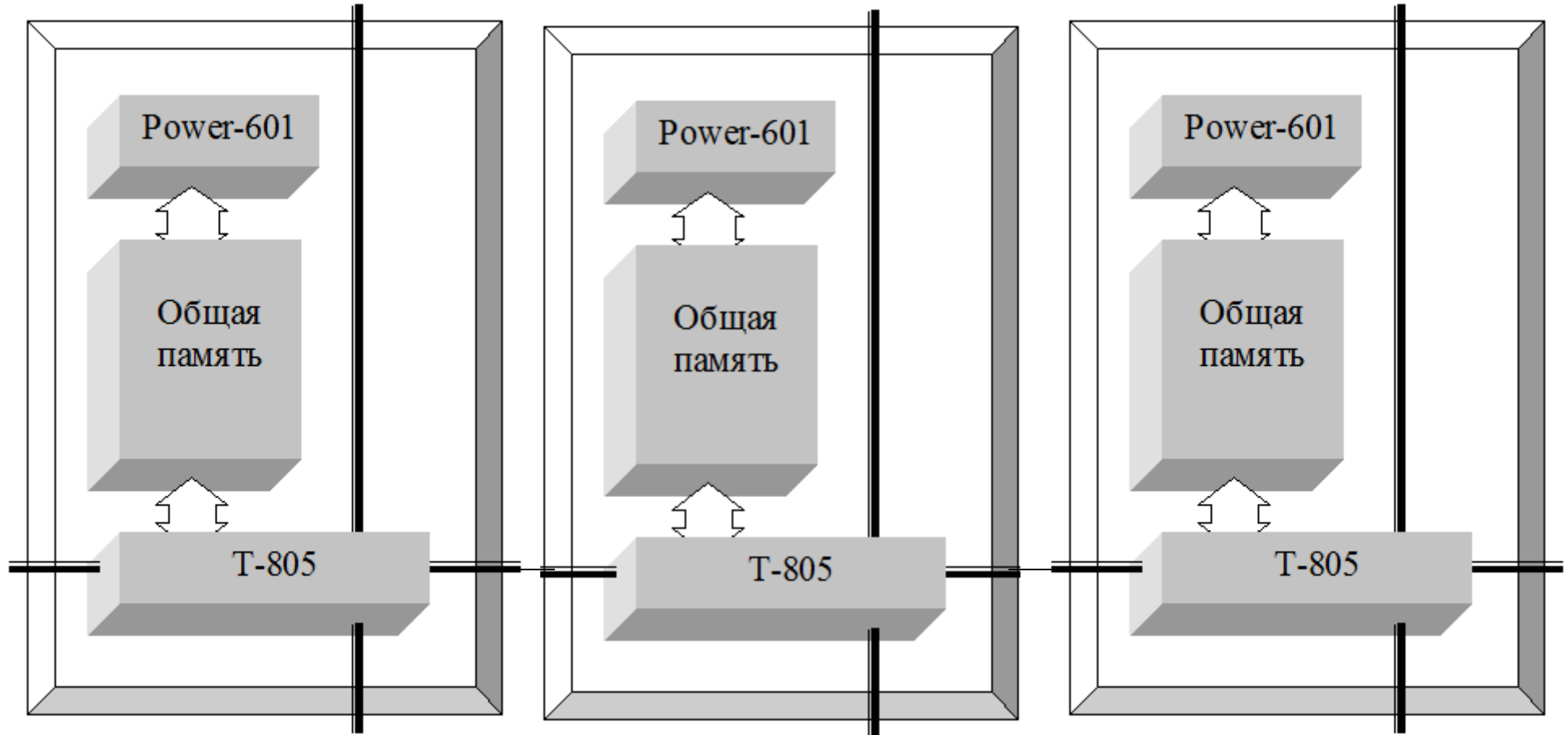
Структура транспьютера Т-800

Узел PowerXplorer



Структура узла PowerXplorer

Гибридная система



Зачем нужны транспьютерные линки?

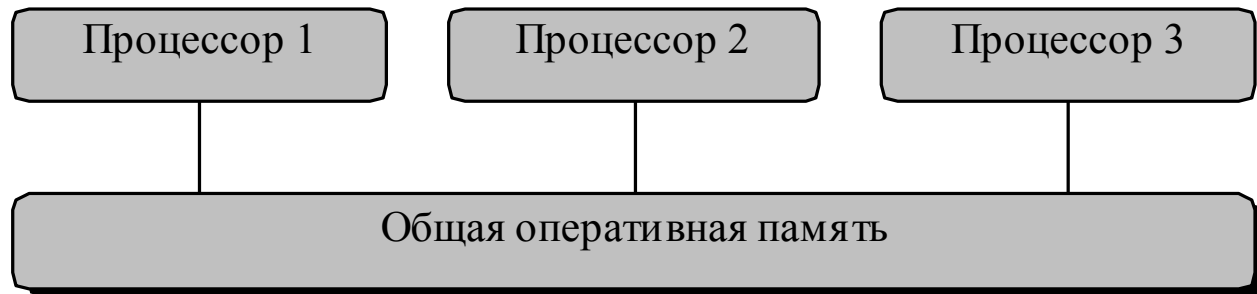
- a. Для ввода-вывода данных
- b. Для подвода питания к транспьютерам
- c. Для передачи данных между транспьютерами

Вопрос

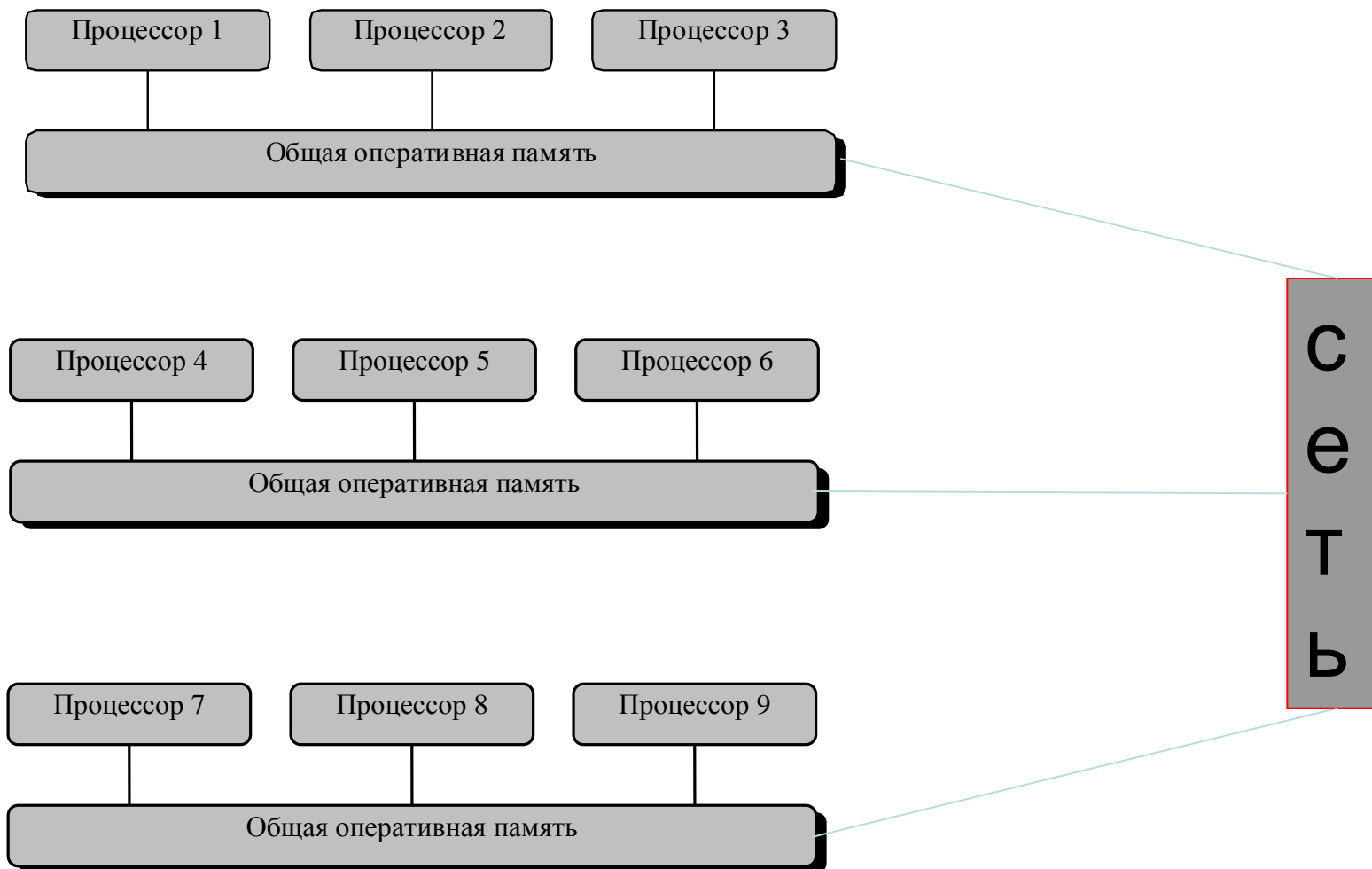
Можно ли на основе транспьютеров делать системы с общей памятью?

- a. Можно делать
- b. Нельзя делать
- c. Только такие системы и можно делать

Многопроцессорные системы с общей памятью

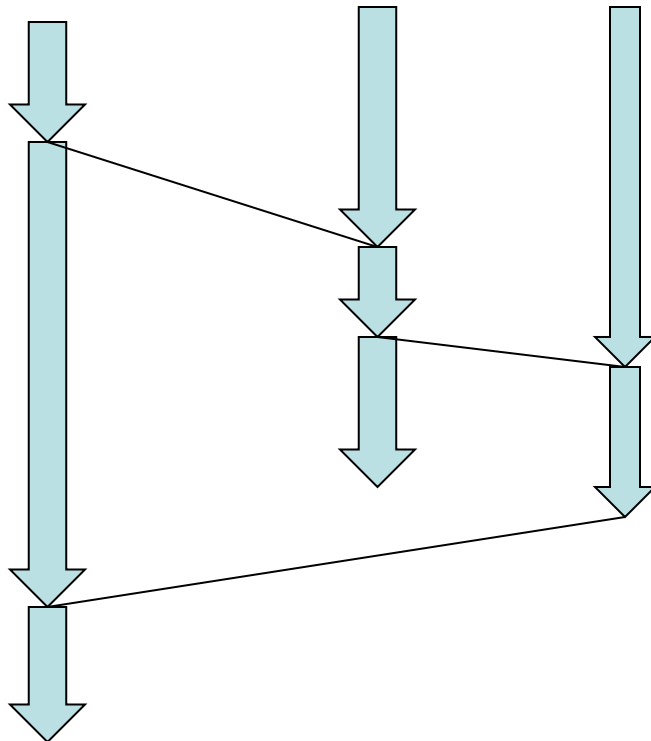


Гибридные вычислительные системы



Уточнение круга рассматриваемых алгоритмов

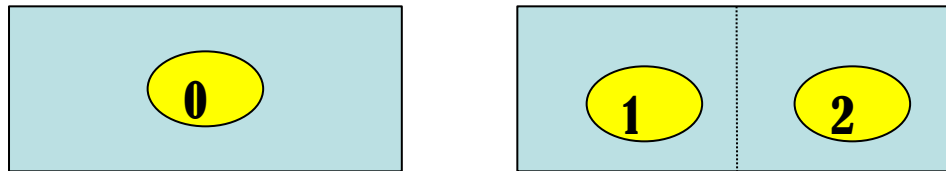
- ❑ Слабо взаимодействующие последовательные процессы



Модель выполнения программы на распределенной памяти

- При запуске указывается число требуемых процессоров N_p и название программы
- На выделенных для расчета узлах запускается N_p копий программы
 - Например, на двух узлах запущены три копии программы. Копия программы с номером 1 не имеет непосредственного доступа к оперативной памяти копий 0 и 2:

Вычислительный узел 1 Вычислительный узел 2



- В каждой копии программы известны значения двух переменных
 - N_p – одинаковое во всех копиях – число копий
 - $rank$ из диапазона $[0 \dots N_p-1]$ – уникальный номер копии
- Любые две копии программы могут непосредственно обмениваться данными с помощью функций передачи сообщений `Send/Recv`

Методы передачи данных

□ Синхронный метод

Send(адрес данных, размер, номер процессора)

Recv(адрес данных, размер, номер процессора)

□ Асинхронные методы

– Небуферизованный

ASend(адрес данных, размер, номер процессора)

ARRecv(адрес данных, размер, номер процессора)

ASync

– Буферизованный

ABSend(адрес данных, размер, номер процессора)

Синхронный

□ A=3

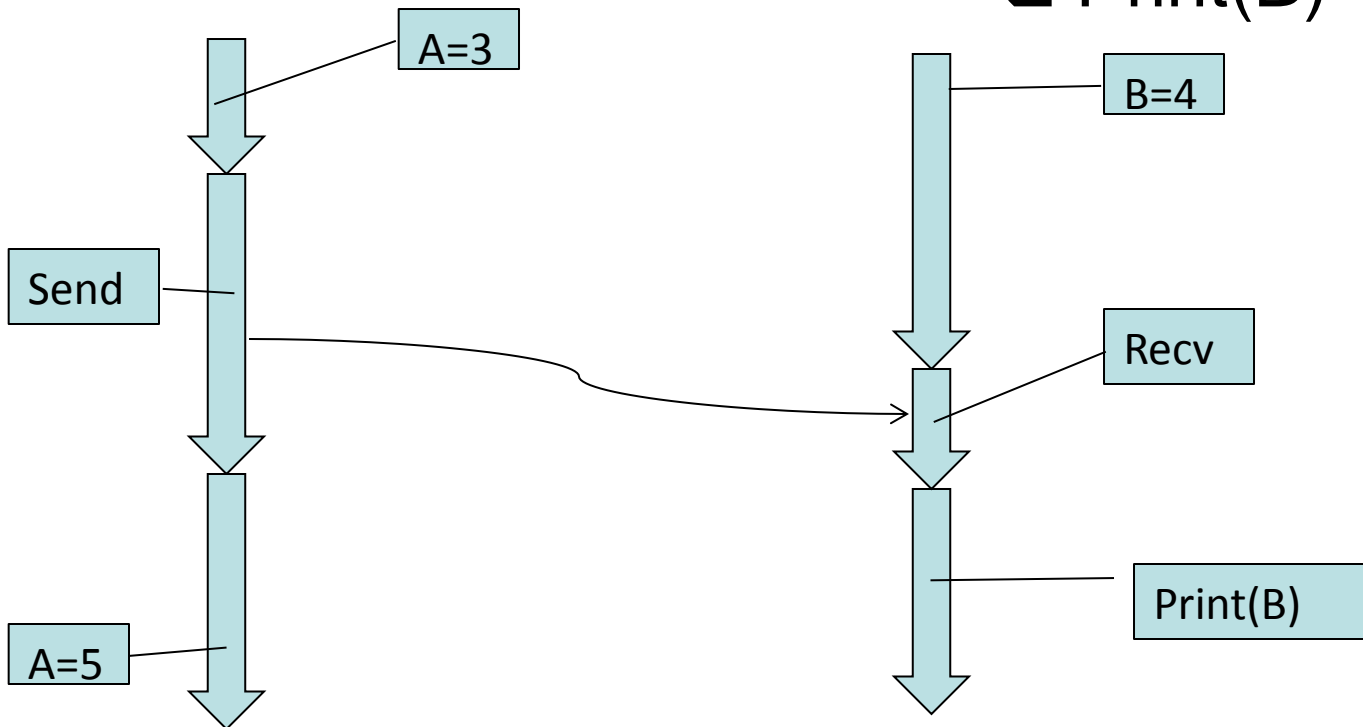
□ Send(&A)

□ A=5

□ B=4

□ Recv(&B)

□ Print(B)

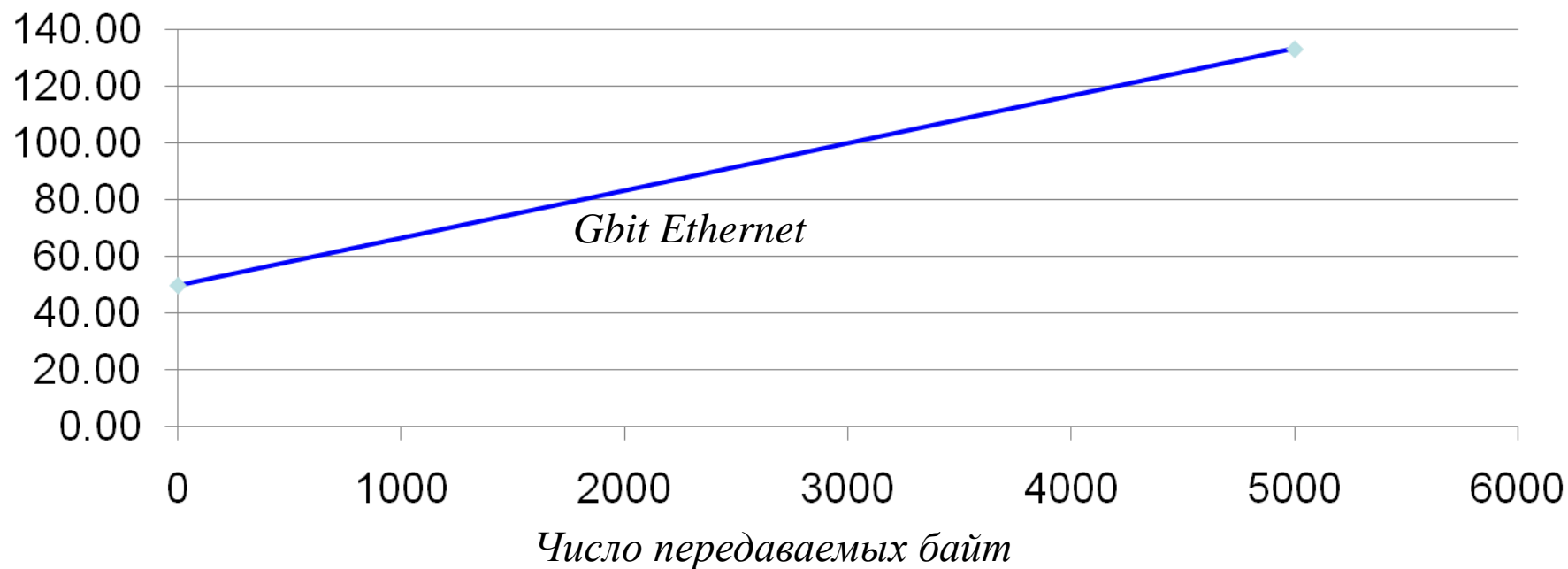


Результат
3

Свойства канала передачи данных

$$T(n) = n * T_{\text{передачи байта}} + T_{\text{латентности}}$$

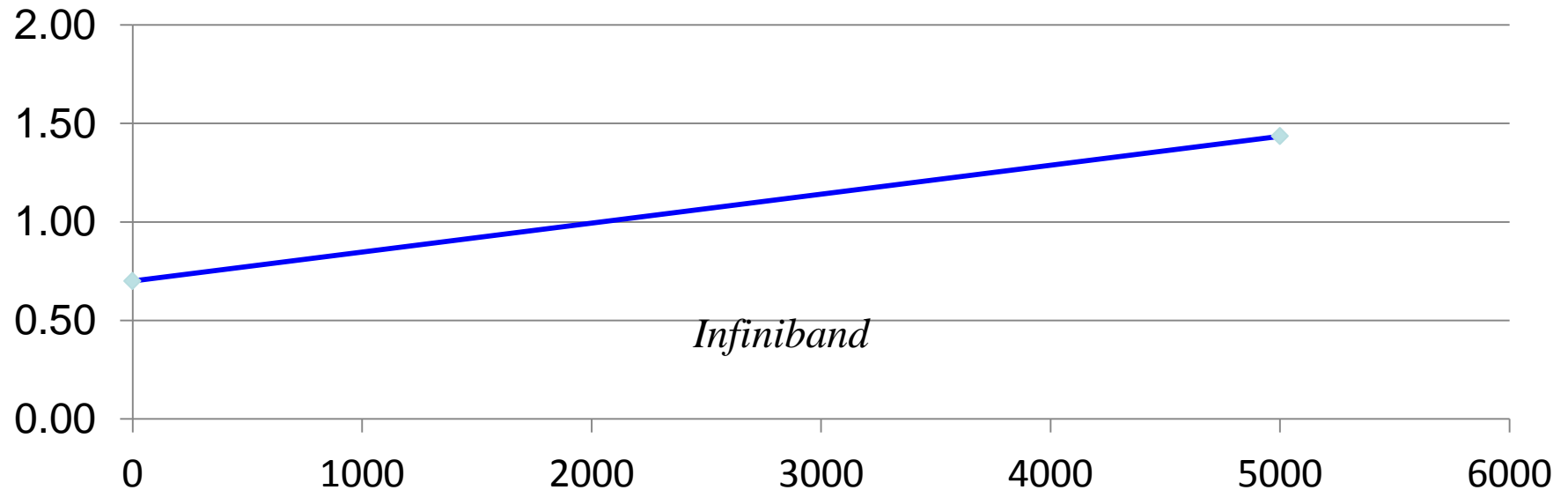
Время передачи данных (мкс)



Свойства канала передачи данных

$$T(n) = n * T_{\text{передачи байта}} + T_{\text{латентности}}$$

Время передачи данных (мкс)



*Число передаваемых байт
6.8 Гбайт/с
За 0.7 мкс передаётся 5 Кбайт*

Вопрос

При каком способе передачи массива чисел будет затрачено меньше времени?

- a. Передача каждого числа отдельным сообщением
- b. Объединение всех чисел в единый массив и передача массива одним сообщением
- c. Не имеет значения

Асинхронные

□ A=3

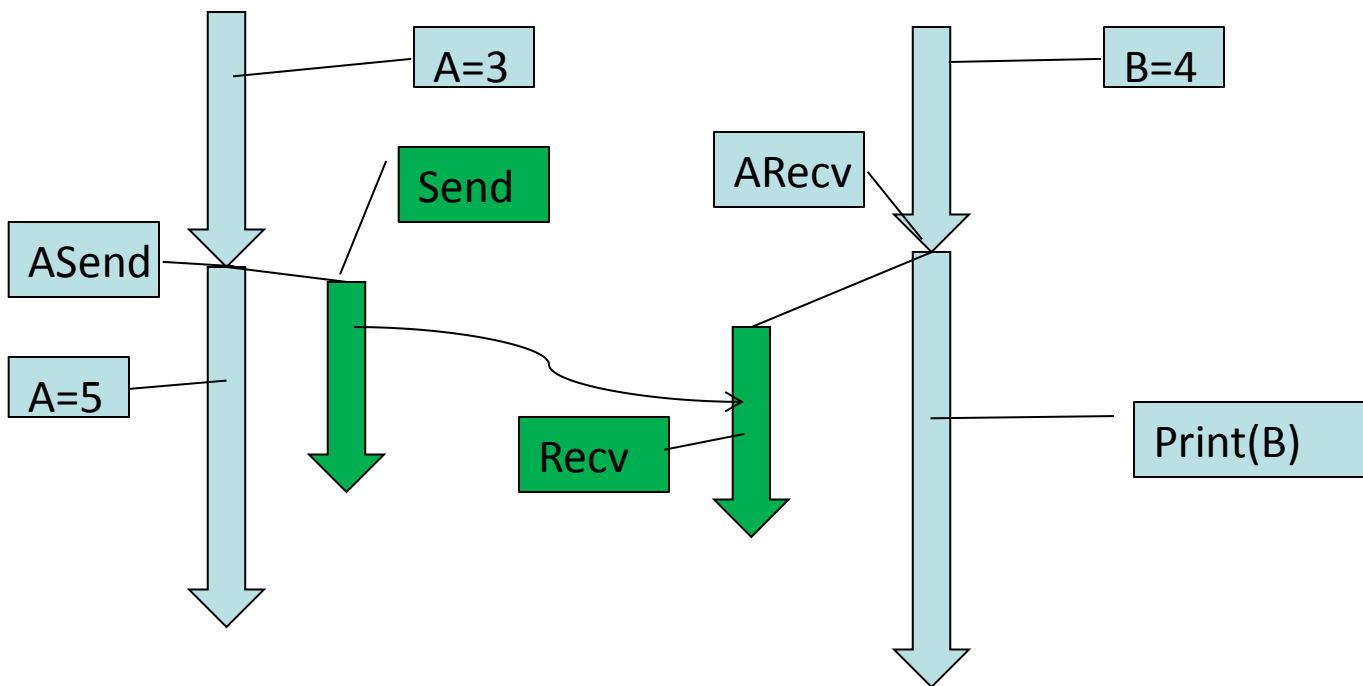
□ ASend(&A)

□ A=5

□ B=4

□ ARecv(&B)

□ Print(B)



Результат

3 ? 4 ? 5

Асинхронные

□ A=3

□ ASend(&A)

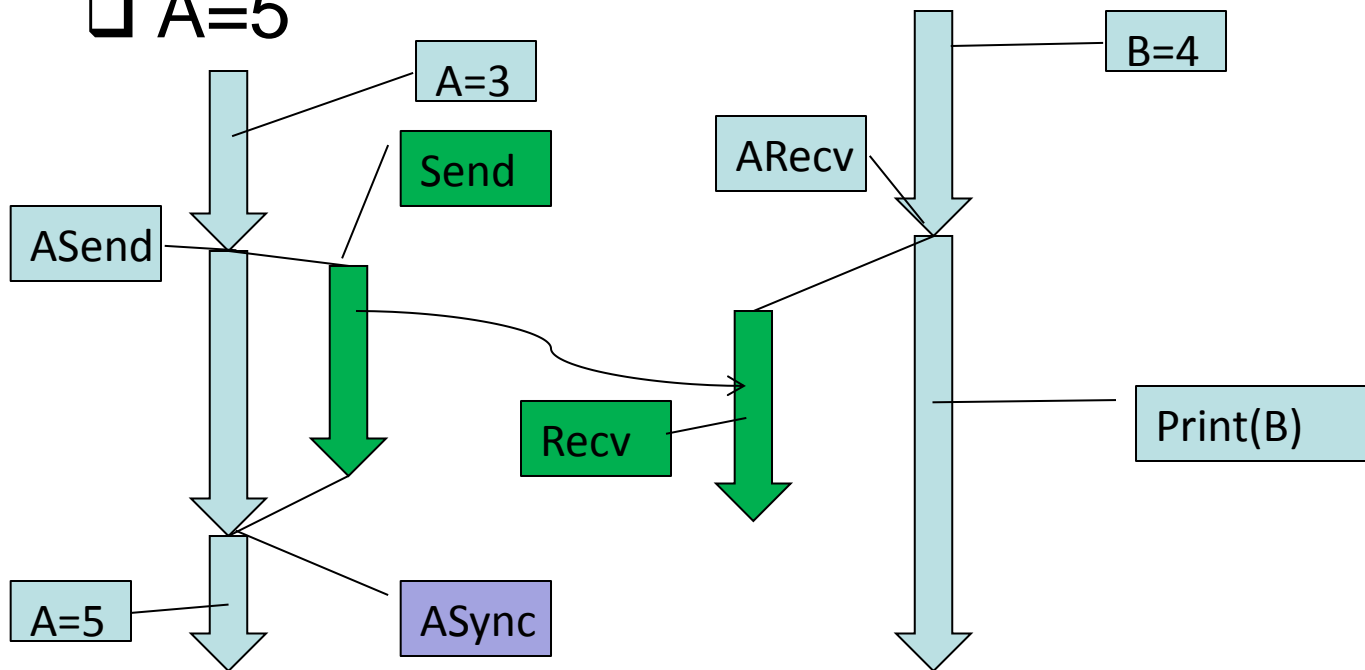
□ Async()

□ A=5

□ B=4

□ ARecv(&B)

□ Print(B)



Результат
3 ? 4

Асинхронные

□ A=3

□ ASend(&A)

□ Async()

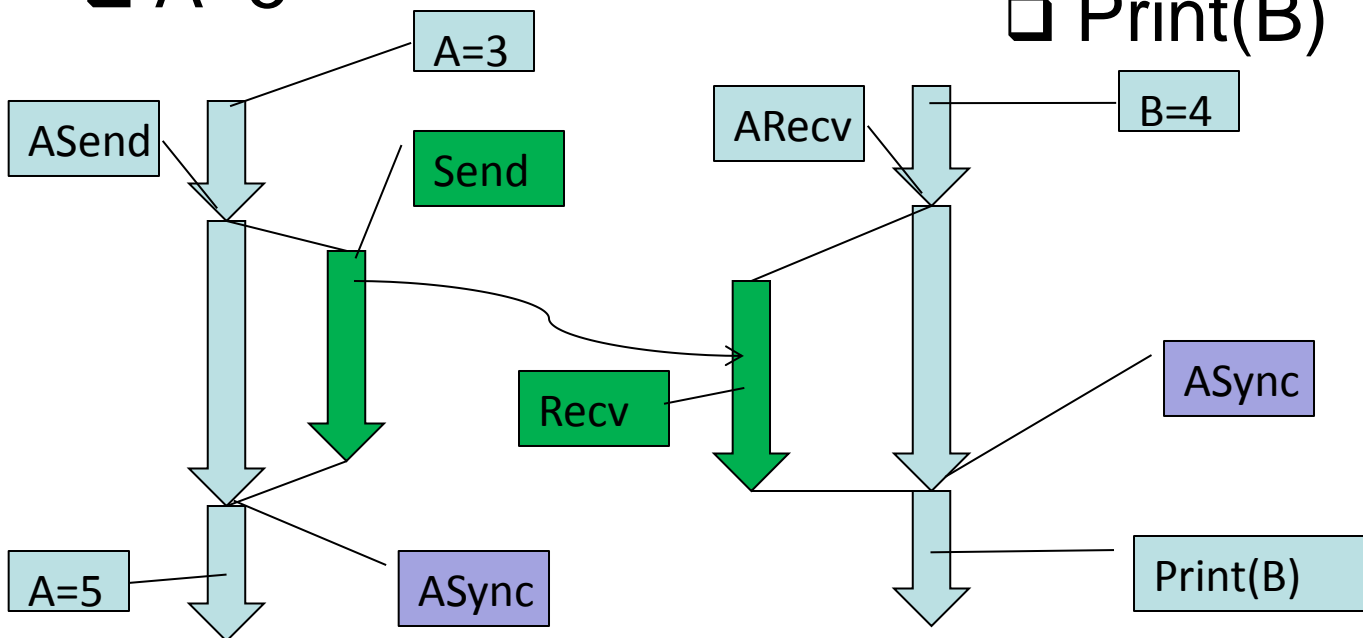
□ A=5

□ B=4

□ ARecv(&B)

□ Async()

□ Print(B)



Результат
3

Асинхронные буферизованные

□ A=3

□ ABSend(&A)

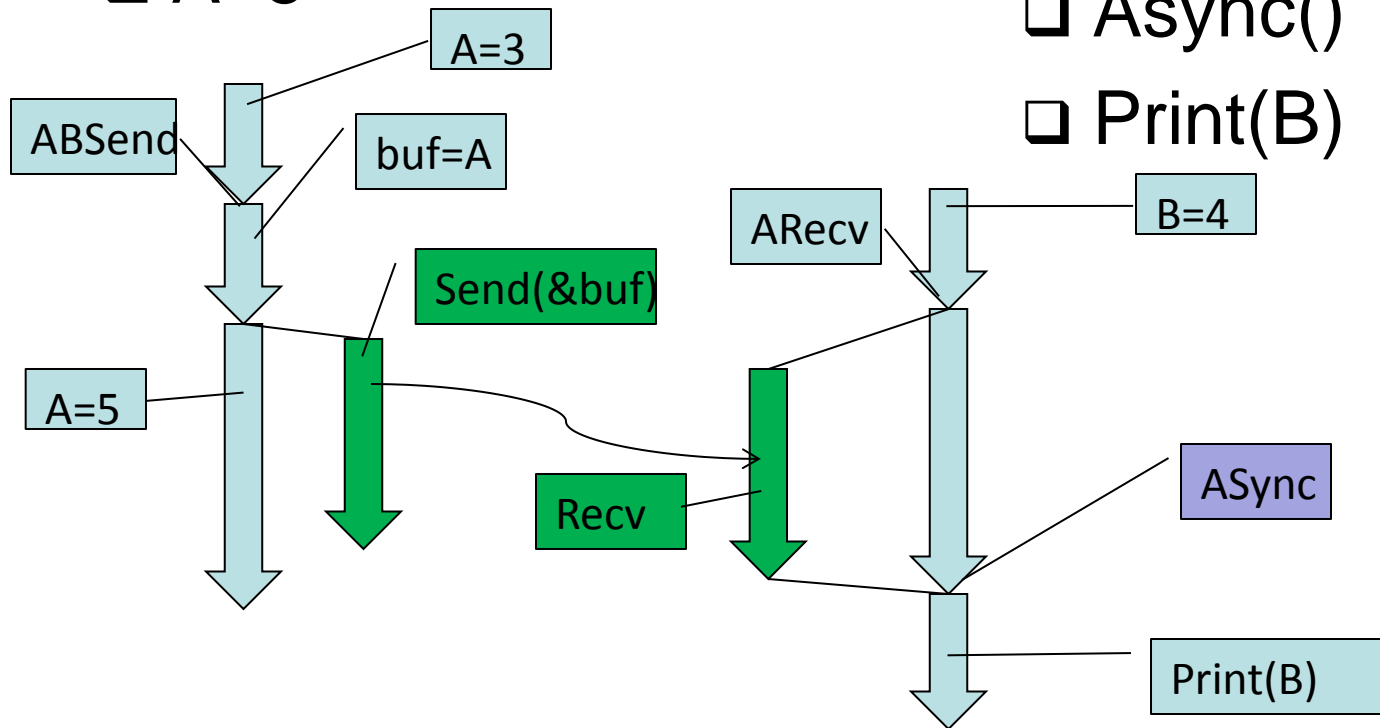
□ A=5

□ B=4

□ ARecv(&B)

□ Async()

□ Print(B)



Результат
3

Методы передачи данных

□ Синхронный метод

Send(адрес данных, размер, номер процессора)

Recv(адрес данных, размер, номер процессора)

□ Асинхронные методы

– Небуферизованный

ASend(адрес данных, размер, номер процессора)

ARRecv(адрес данных, размер, номер процессора)

ASync

– Буферизованный

ABSend(адрес данных, размер, номер процессора)

Контакты

Якобовский М.В., чл.-корр. РАН, проф., д.ф.-м.н.,
Заместитель директора по научной работе
Института прикладной математики
им. М.В.Келдыша Российской академии наук

[mail: lira@imamod.ru](mailto:lira@imamod.ru)

[web: http://lira.imamod.ru](http://lira.imamod.ru)