

*Московский государственный университет имени М.В.Ломоносова
Факультет Вычислительной математики и кибернетики*



СУПЕРКОМПЬЮТЕРЫ И ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

(часть 3)

Вл.В.Воеводин

**Зав.кафедрой Суперкомпьютеров и квантовой информатики ВМК МГУ,
Директор НИВЦ МГУ,
чл.-корр. РАН, д.ф.-м.н., профессор**

voevodin@parallel.ru

ВМК МГУ, 2020

Важные сокращения

Мега (Mega)	$- 10^6$	(миллион)
Гига (Giga)	$- 10^9$	(биллион / миллиард)
Тера (Tera)	$- 10^{12}$	(триллион)
Пета (Peta)	$- 10^{15}$	(квадриллион)
Экза (Exa)	$- 10^{18}$	(квинтиллион)

Флоп/с, *Flop/s* – *F*loating point *o*perations
per *s*econd

15 Tflop/s = $15 * 10^{12}$ арифметических операций
в секунду над вещественными числами,
представленными в форме с плавающей точкой.

Суперкомпьютер Fujitsu Fugaku, Япония (#1 Top500 в 2020 г.)



158 976 выч.узлов, 432 стойки

CPU: A64FX ARM v8.2-A, 48/52 cores
Векторные инструкции (SVE),
аппаратная поддержка барьеров...
Int: 1,2,4,8 байт; Float: 16,32,64 бит

Производительность:
Пик (теория): 513 Pflop/s
Тест Linpack: 415 Pflop/s (81%)
Оперативная память = 4.85 Пбайт
Энергопотребление = 28 МВт
Площадь = 1920 м²

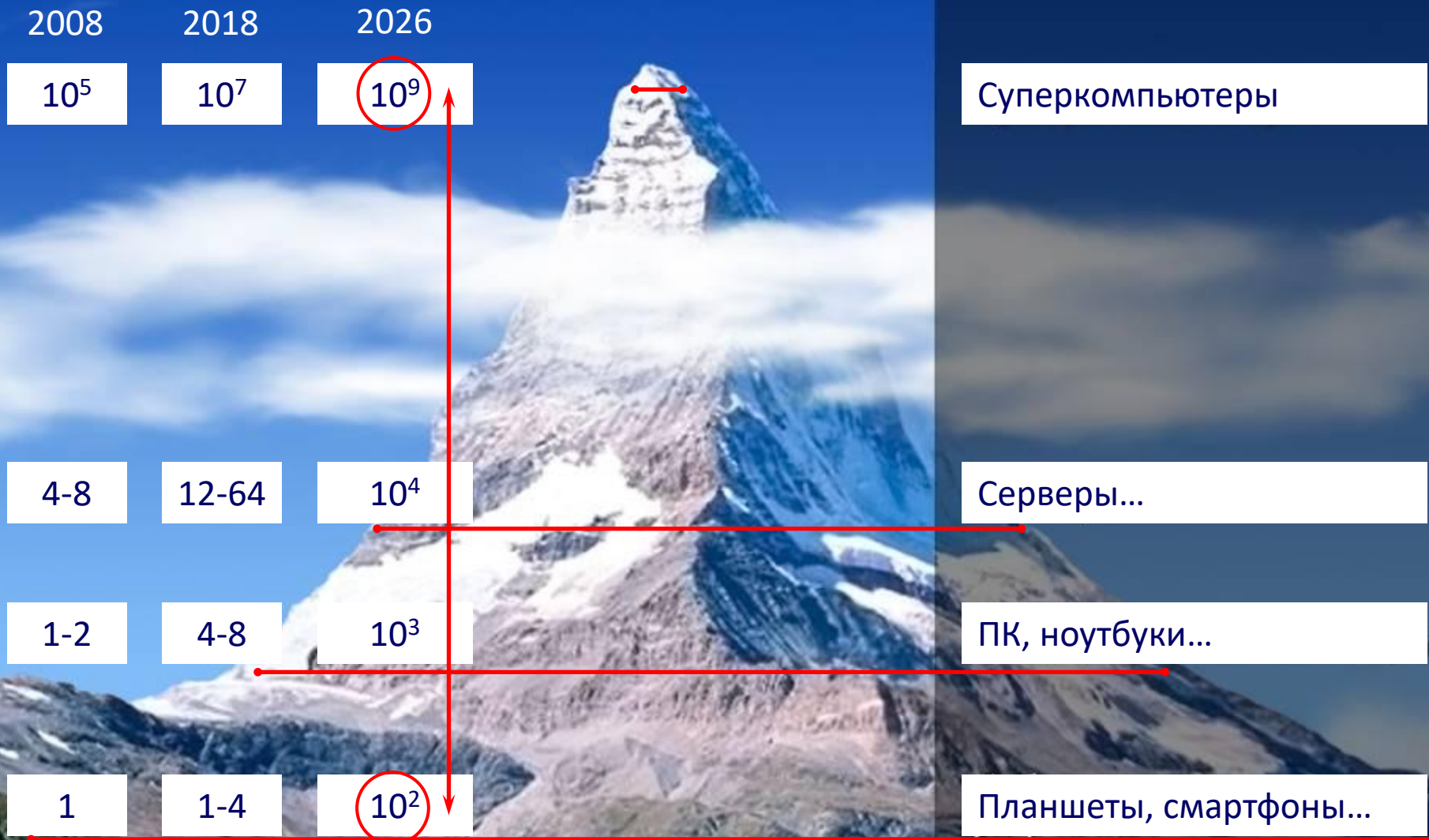
Суперкомпьютеры... Зачем?

Неужели есть настолько **сложные задачи**, что для их решения хорошего сервера не хватает ?

Неужели есть настолько **важные задачи**, которые оправдывают крайне высокую стоимость суперкомпьютеров ?

Параллельный компьютерный мир

Степень параллелизма

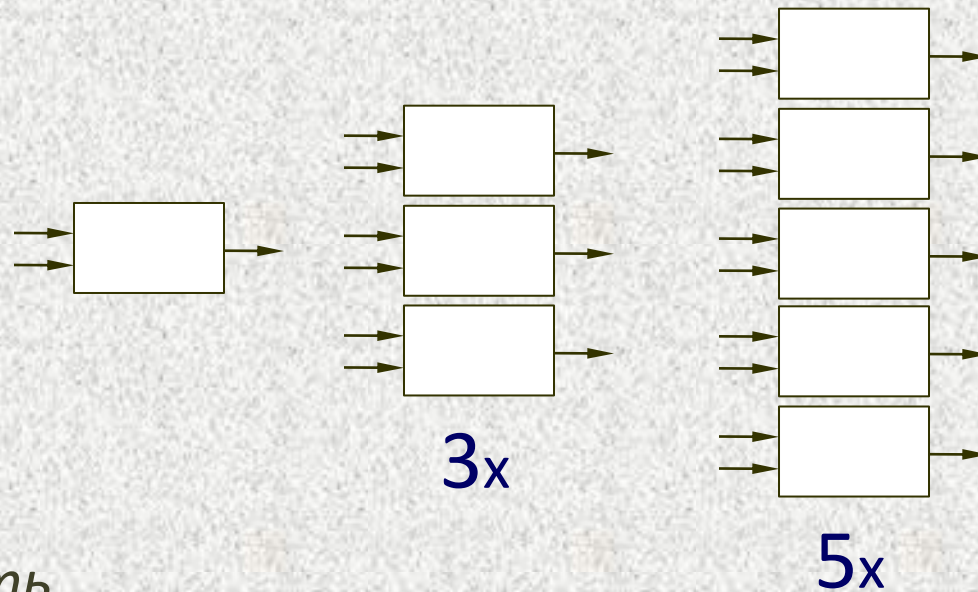


Два вида параллельной обработки данных

- *Параллелизм*
- *Конвейерность*

Параллелизм в архитектуре компьютеров

- Параллелизм



- Конвейерность



Длина конвейера = число ступеней конвейера.

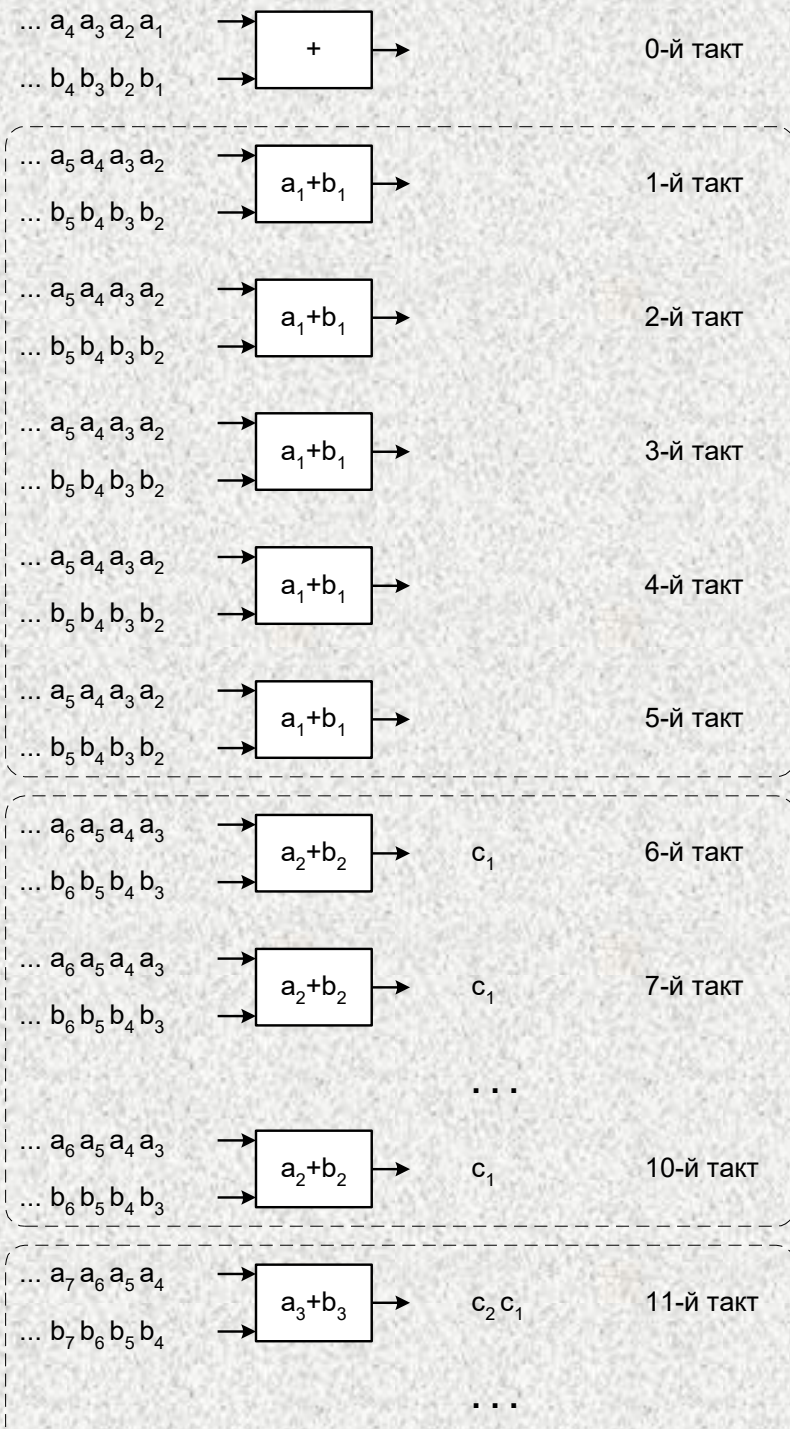
Последовательная обработка данных

Суммирование векторов $C = A + B$ с помощью **1** последовательного устройства.

Устройство выполняет **1** операцию за **5** тактов.

Векторы A и B содержат по **100** элементов.

Время выполнения данной операции: **500** тактов.



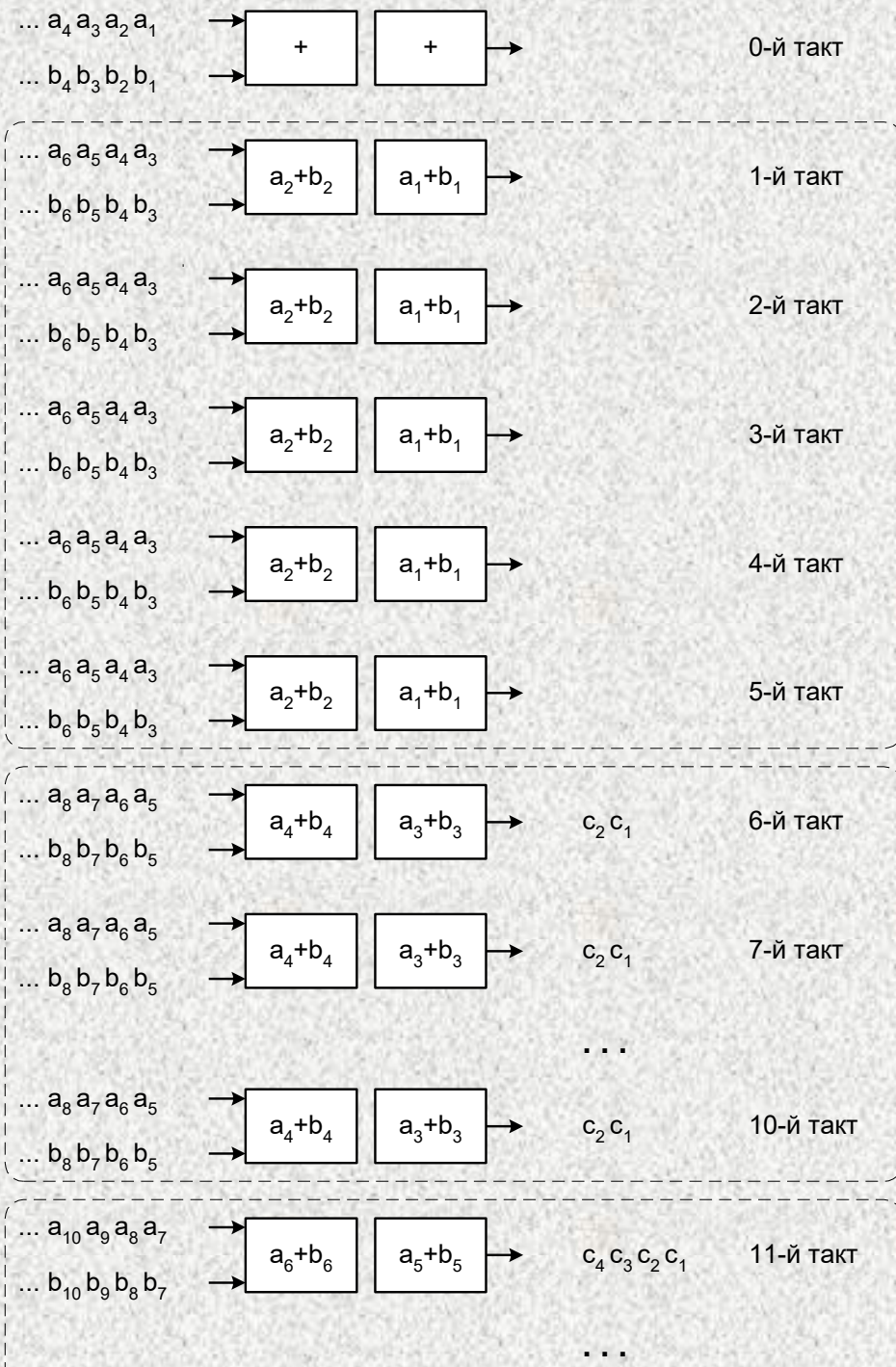
Параллельная обработка данных

Суммирование векторов $C = A + B$
с помощью **2** одинаковых
последовательных устройств.

Каждое устройство выполняет по
1 операции за **5** тактов.

Векторы A и B содержат по
100 элементов.

Время выполнения данной
операции: **250** тактов.



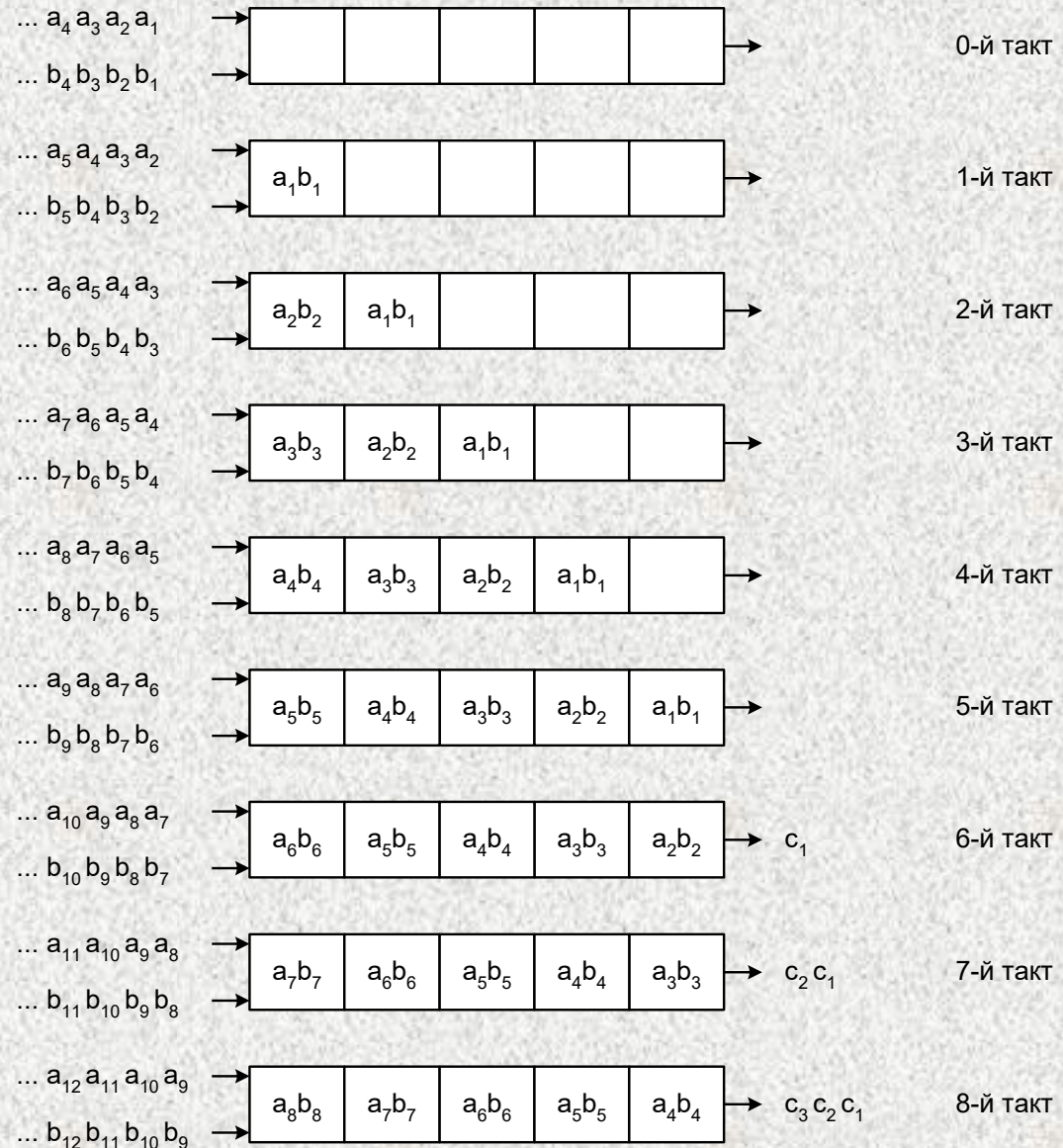
Конвейерная обработка данных

Суммирование векторов
 $C = A + B$ с помощью
конвейерного устройства.

Конвейерное устройство
состоит из **5** ступеней.
Каждая из **5** ступеней
срабатывает за **1** такт.

Векторы A и B содержат по
100 элементов.

Время выполнения данной
операции: **104** такта.



Конвейерная обработка данных (основные понятия)



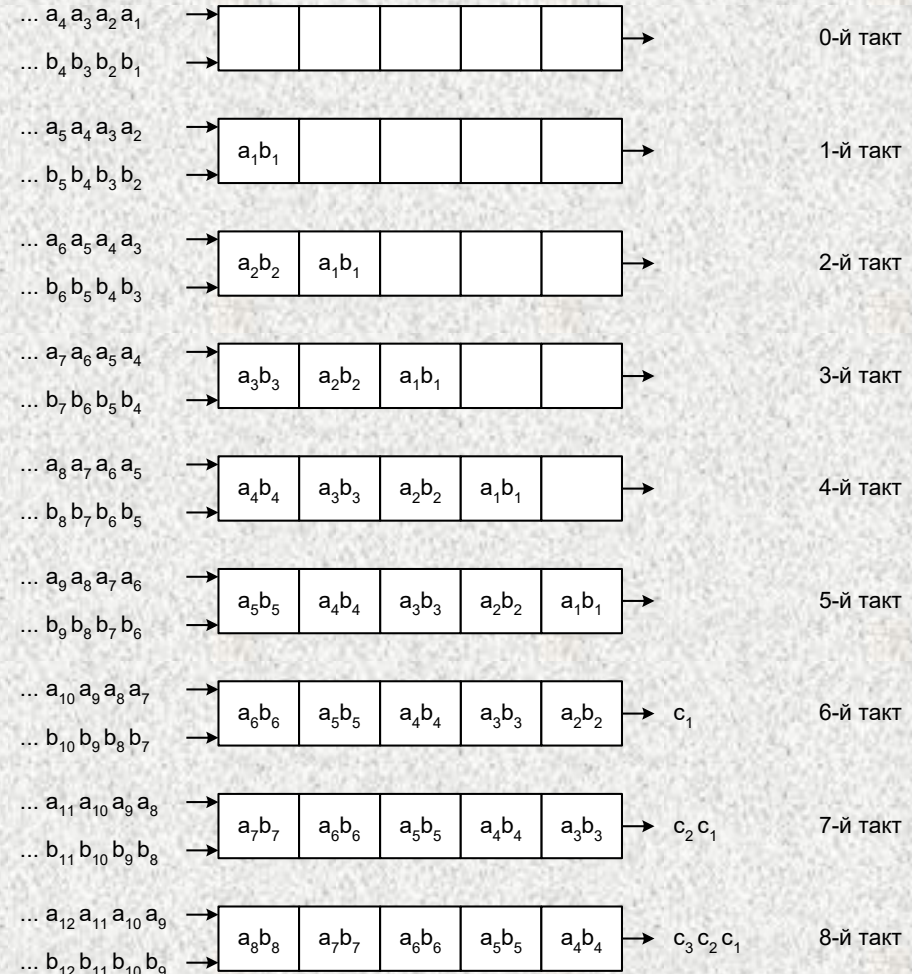
L - длина конвейера (число ступеней конвейера).
Каждая ступень срабатывает за 1 такт.
Темп выдачи результатов – каждый такт.

Время обработки всего набора
из N входных данных:

$$T = L + (N - 1)$$

Время (число тактов) для
заполнения всего
конвейера
(одновременно будет
обработан первый
входной аргумент).

Время (число тактов) для
обработки оставшегося
набора из $N-1$ входных
аргументов.



Конвейерная обработка данных (вопрос...)

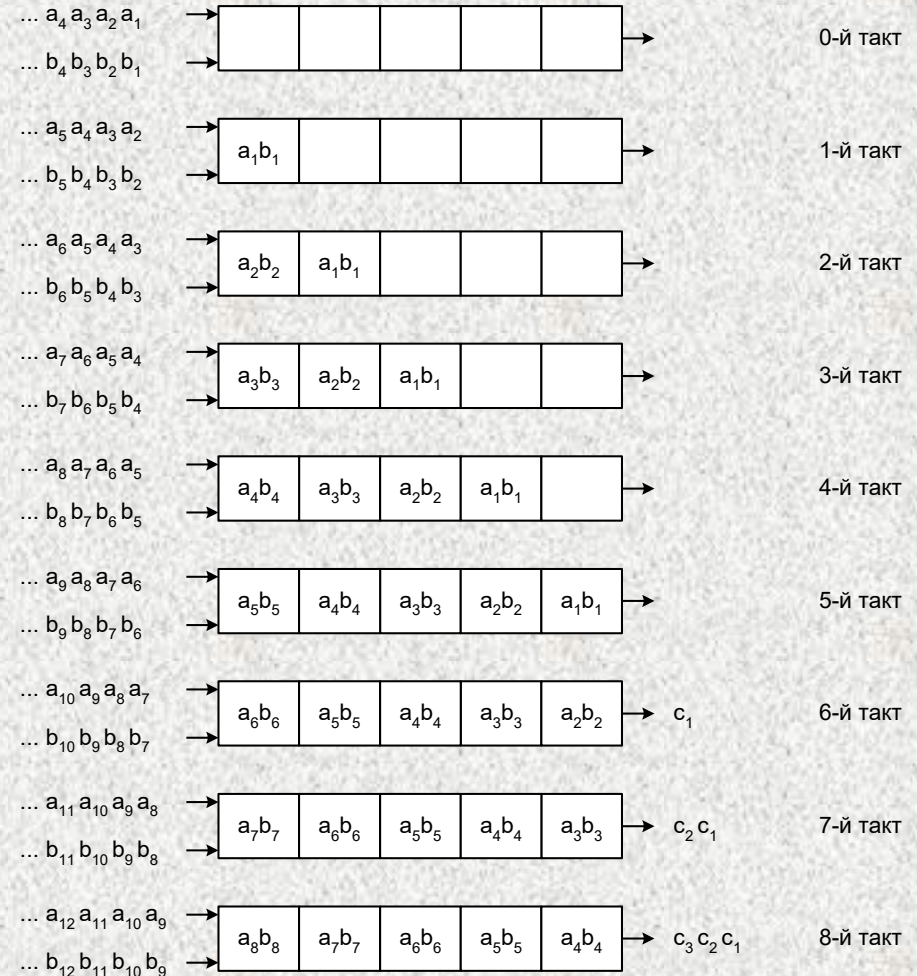


L - длина конвейера (число ступеней конвейера).

Каждая ступень срабатывает за T_i тактов.

С какой частотой будут выдаваться результаты
в устоявшемся режиме?

$$\max_i (T_i)$$



Конвейерная обработка данных

(зацепление устройств – это часто на практике!)



L_1 - длина конвейера сложения.

L_2 - длина конвейера умножения.

Каждая ступень срабатывает за 1 такт.

Нужно выполнить операцию

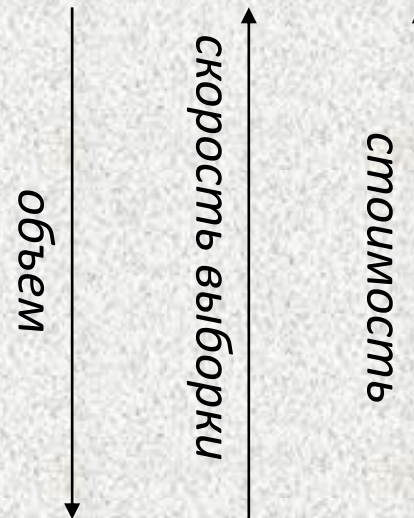
$$A_i = B_i + C_i * d, i=1..N$$

Зацепление функциональных устройств – это такой режим обработки, при котором выход одного функционального устройства подаётся на вход другого.

- Время обработки в режиме без зацепления: $T = L_2 + (N-1) + L_1 + (N-1)$
- Время обработки в режиме с зацеплением: $T = L_2 + L_1 + (N-1)$

Иерархия памяти

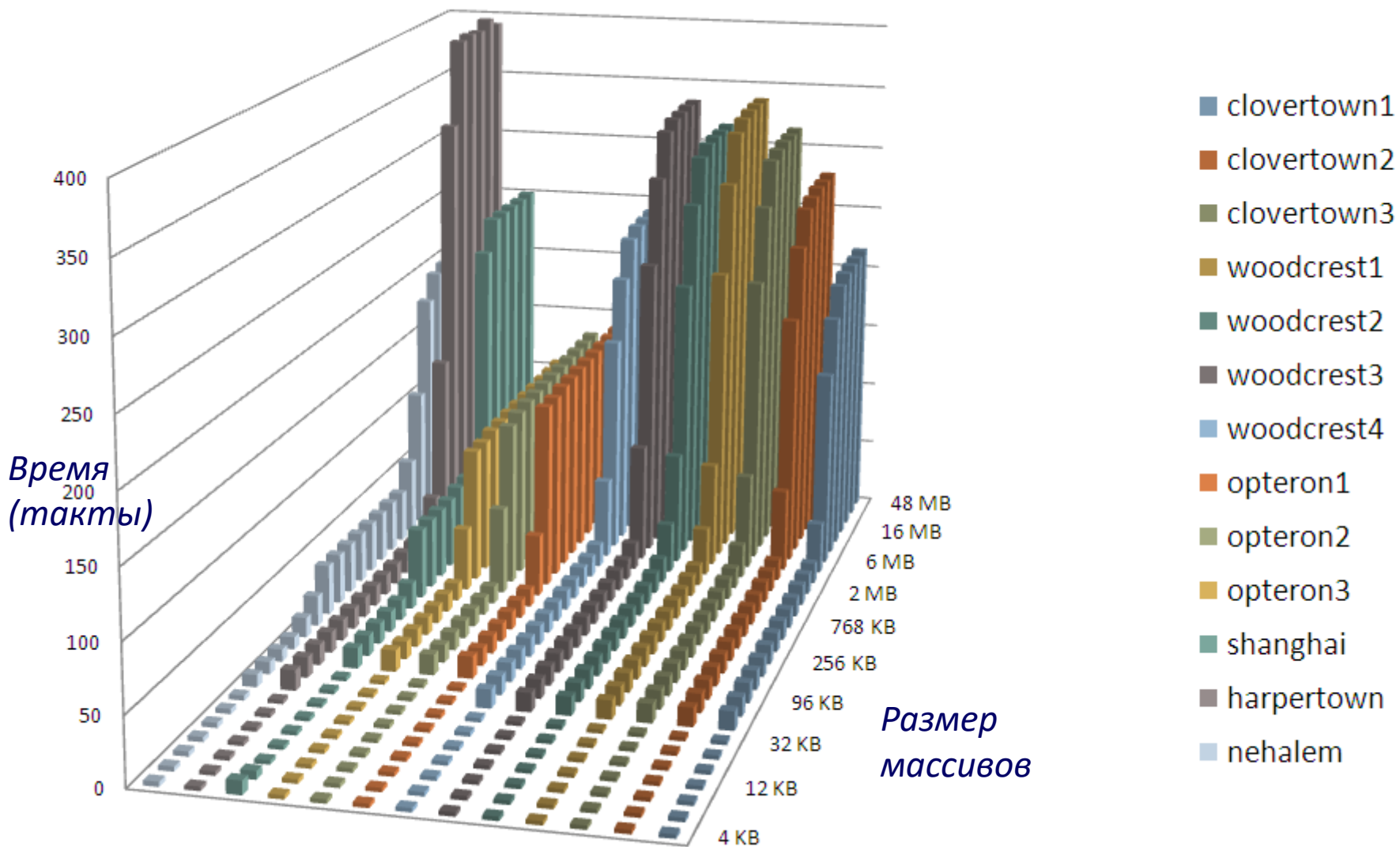
- регистры
- кэш-память 1-го уровня
- кэш-память 2-го уровня
- оперативная память
- дисковые устройства
- ленточные накопители
- ...



Очень важные характеристики программ:

- Локальность вычислений
- Локальность использования данных

Время чтения для различных уровней иерархии памяти



Свойства работы программ с памятью

Ключевое понятие – **локальность**: насколько близко расположены данные/команды друг относительно друга.

- Локальность данных,
- Локальность команд.
- Пространственная локальность данных. Отражает среднее расстояние по памяти между запрашиваемыми данными.
- Временная локальность данных. Показывает среднюю частоту обращений по одному адресу в память за время исполнения программы.

*Суперкомпьютеры –
это колоссальная мощь.
Но есть ли здесь проблемы ?*

*Увы, да, эффективная организация параллельных
вычислений –
это очень не просто...*

Производительность компьютера и время решения задачи

Вскапываем огород, 10×10 м – все хорошо: если нужно ускориться, то зовем 5, 10, 50 помощников...

Копаем яму, $1 \times 1 \times 1$ м – как ускорить процесс? Еще 10 или 50 помощников сильно время не уменьшат...

Копаем фундамент: разметку может делать только бригадир, остальное – рабочие.

- 1 час разметка, остальное - 10 часов (1 рабочий) = 11 часов
- 1 час разметка, остальное - 1 час (10 рабочих) = 2 часа
- 1 час разметка, остальное - 6 мин. (100 рабочих) = 1 час 6 мин.

...бригадир становится “узким местом”.

Характеристики параллельных программ (основные обозначения)

f – доля последовательных операций ($0 \leq f \leq 1$),

p – число процессоров (ядер, вычислительных узлов),

T_1 – время работы программы на одном процессоре,

T_p – время работы программы на системе из p процессоров,

$S = \frac{T_1}{T_p}$ – это ускорение работы программы при переходе с одного процессора на систему из p процессоров.

Закон Амдала

(формулируется просто, но влияет сильно)

p – число процессоров (ядер, вычислительных узлов),

T_1 – время работы программы на одном процессоре,

T_p – время работы программы на системе из p процессоров,

$S = \frac{T_1}{T_p}$ – это ускорение работы программы при переходе с одного процессора на систему из p процессоров (во сколько раз программа начинает работать быстрее),

f – доля последовательных операций в исходной программе ($0 \leq f \leq 1$).

Закон Амдала.

Ускорение работы программы при переходе с одного процессора на систему из p процессоров можно оценить следующим образом:

$$S \leq \frac{1}{f + (1 - f) / p}$$

На практике: $S < p$; идеальный случай: $S = p$ – линейное ускорение работы программы.

Закон Амдала. Следствие

$$S \approx \frac{1}{f} \quad (\text{при большом числе процессоров})$$

На практике. Если доля последовательных операций в некоторой программе равна 0.1, значит **вне зависимости от числа используемых процессоров** ускорение не превысит 10.

Закон Амдала. Следствие

В теории. Для того чтобы ускорить программу в q раз, необходимо ускорить не менее, чем в q раз не менее, чем $(1-1/q)$ -ю часть программы.

На практике. Нужно ускорить работу программы в 100 раз. Значит необходимо ускорить не менее, чем в 100 раз не менее, чем 99% этой программы.

Производительность компьютера

(теория и практика)

- Пиковая производительность компьютера, R_{peak}

Определяется по архитектуре компьютера в предположении, что все устройства работают с полной загрузкой, без остановок, необходимые аргументы всегда готовы, а передача данных выполняется мгновенно. Это теоретически максимальная производительность компьютера.

- Реальная производительность компьютера, R_{max}

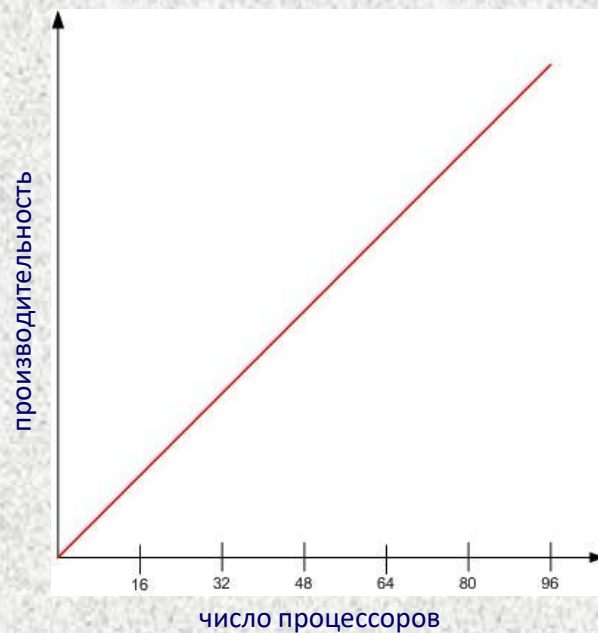
Индивидуальна для каждой программы и/или каждого компьютера. Определяется по параметрам работы программы на компьютере:

$$R_{max} = \frac{\text{число операций в программе}}{\text{время выполнения программы на компьютере}}$$

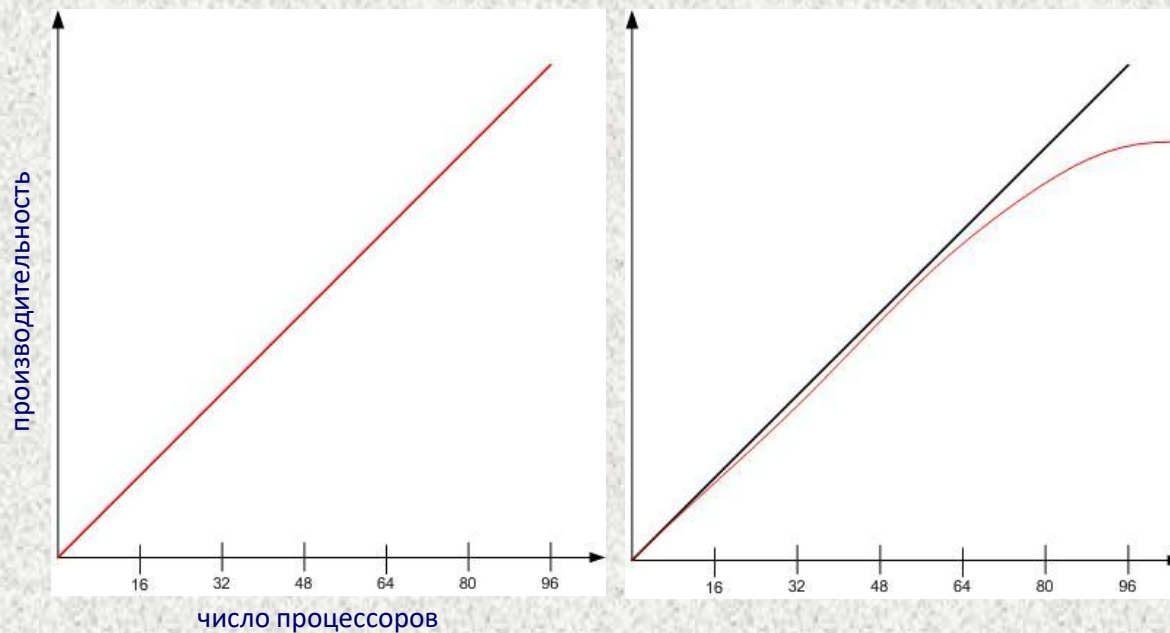
Пиковая производительность
недостижима на практике

$$\Rightarrow R_{max} < R_{peak}$$

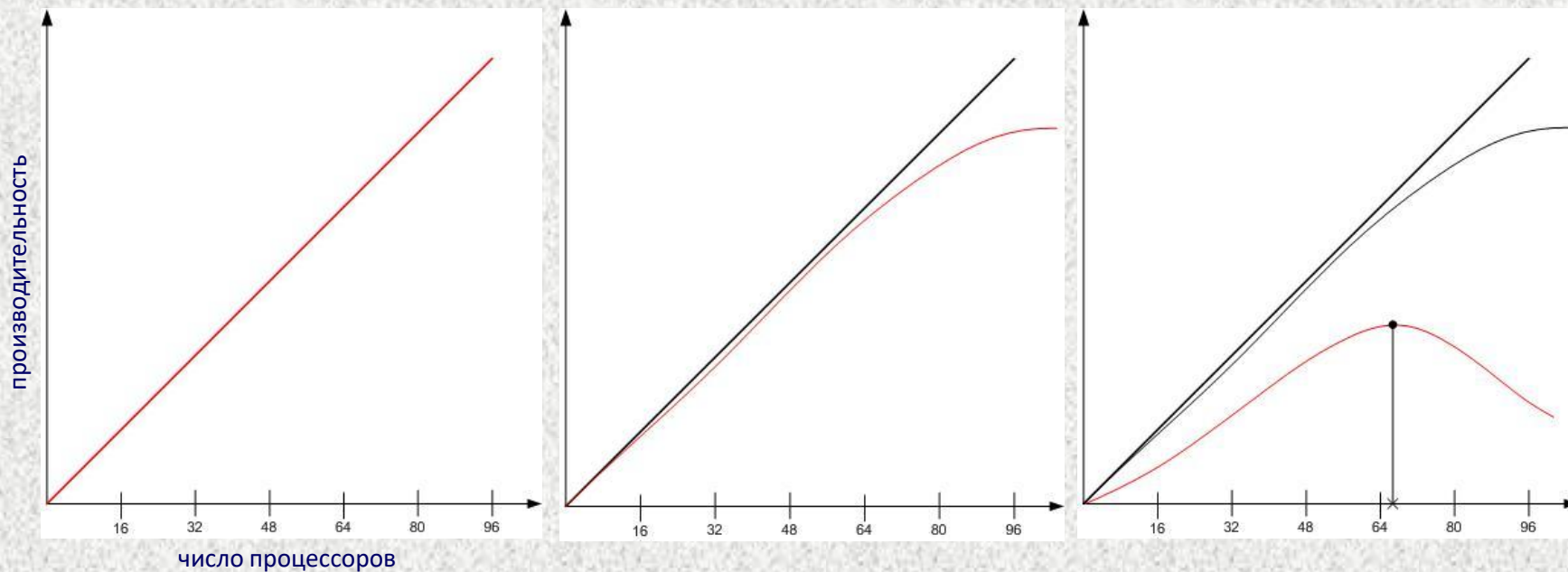
Рост производительности параллельных вычислительных систем (в теории)



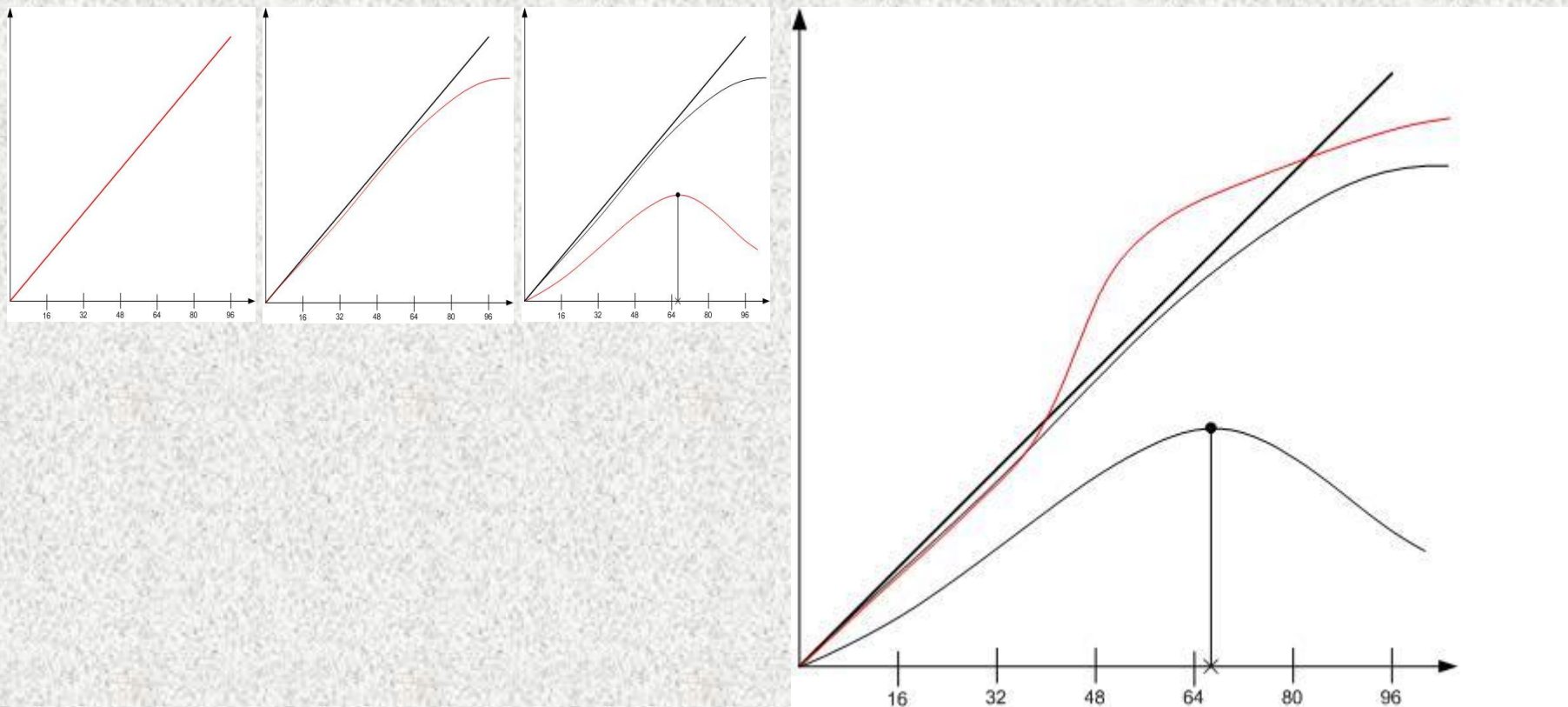
Рост производительности параллельных вычислительных систем (в теории и на практике)



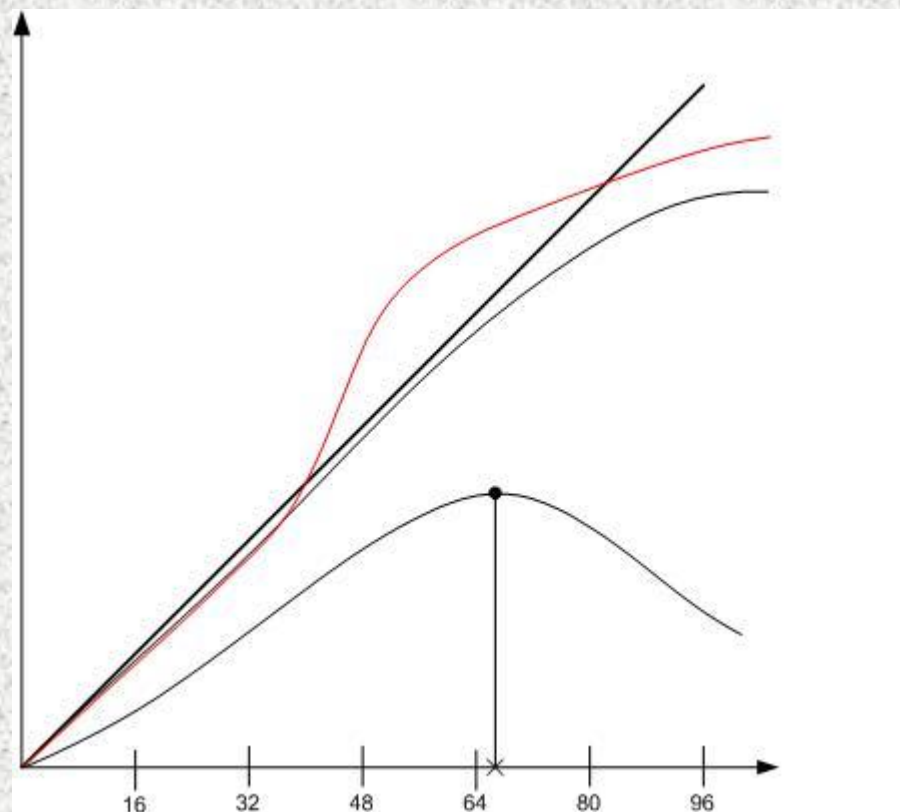
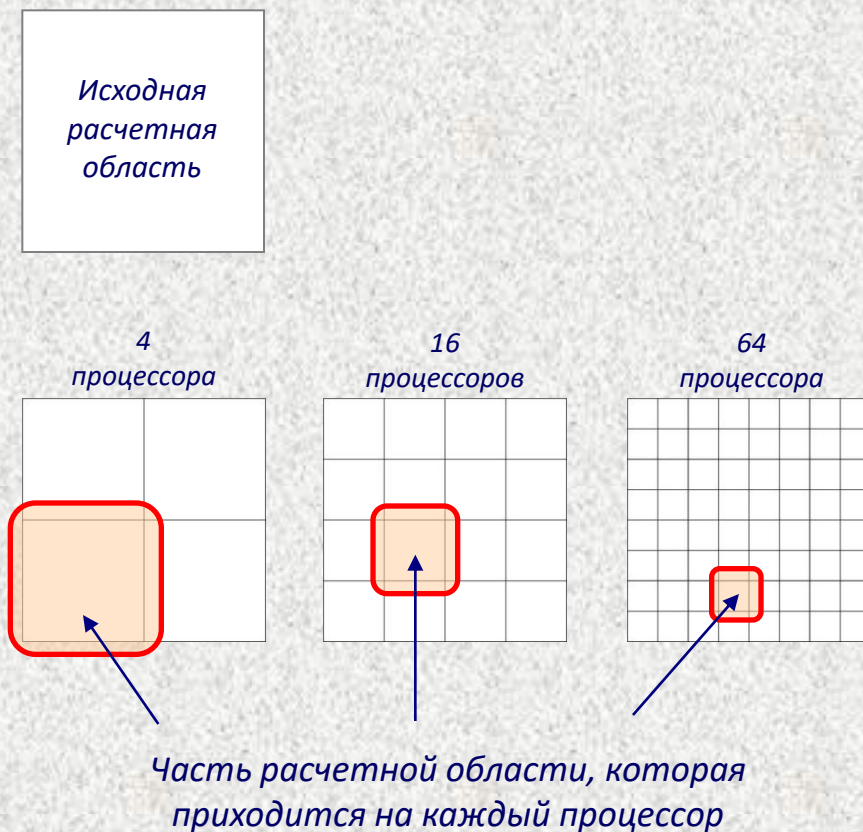
Рост производительности параллельных вычислительных систем (в теории и на практике)



Рост производительности параллельных вычислительных систем (в теории и на практике)



Рост производительности параллельных вычислительных систем (в теории и на практике, **суперлинейное ускорение**)



$S > p$ – суперлинейное ускорение работы программы, эффект возникает из-за кэш-памяти.

Эффективность работы компьютера (программы)

(практика)

- Пиковая производительность компьютера, R_{peak}
- Реальная производительность компьютера, R_{max}

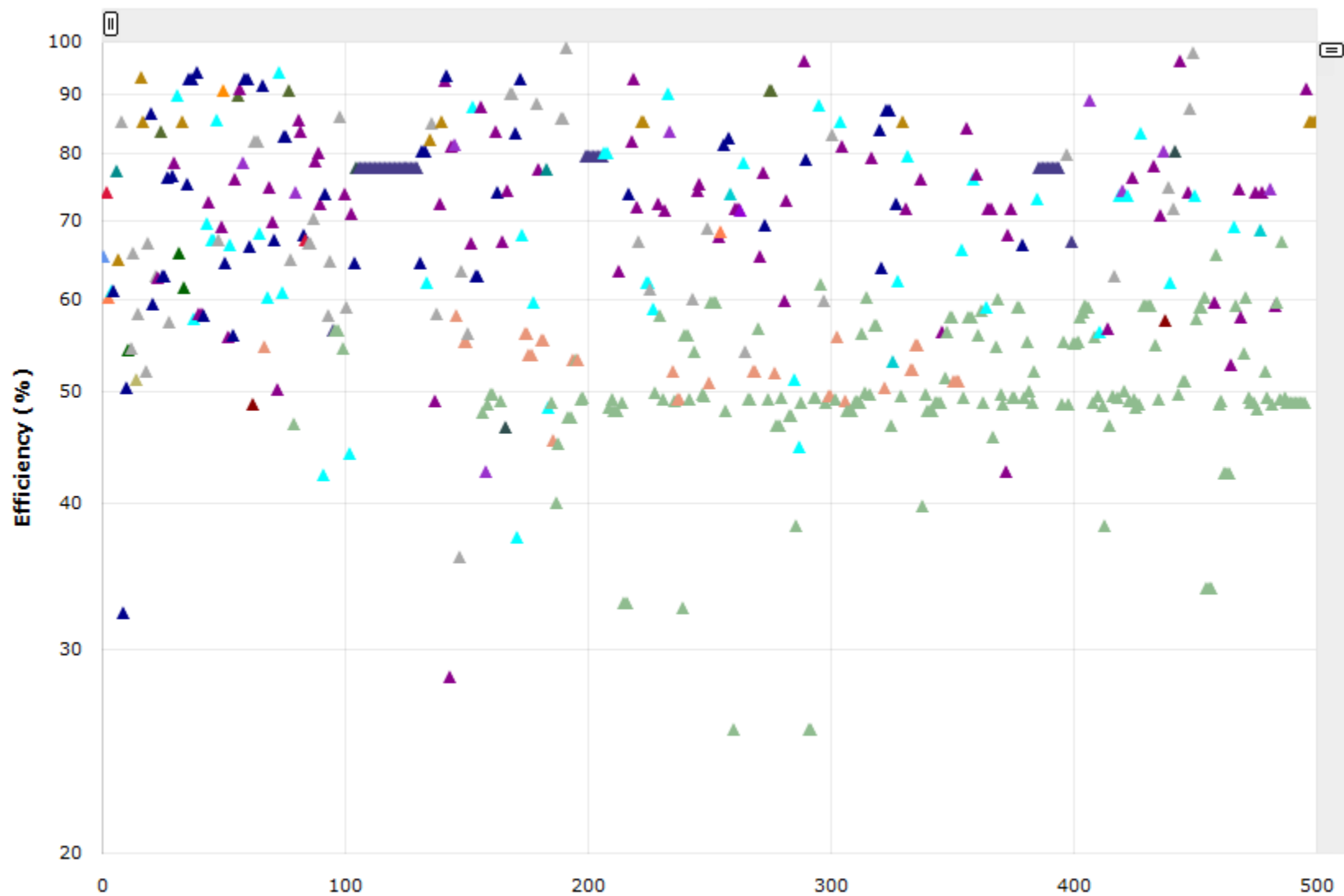
$$\text{Эффективность} = \frac{R_{max}}{R_{peak}}$$

- эффективность реализации программы;
- эффективность работы компьютера (КПД);

Эффективность показывает, в какой степени (насколько полно) используются вычислительные возможности компьютера.

На тестах и простых примерах: $R_{max} < R_{peak}$; В реальности: $R_{max} \ll R_{peak}$.

Топ500, Linpack, Эффективность



Параллельные вычисления: важны все детали...

Вопрос.

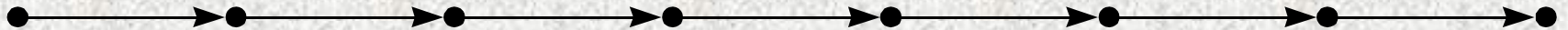
*Трое мужчин могут покрасить три забора за три часа.
За сколько времени покрасит один человек один забор?*

Ответ:

Возможен любой промежуток времени, как сколь угодно маленький, так и сколь угодно большой...

Суммирование элементов массива. Параллелизма нет...

```
s = 0.0;  
for ( i = 0; i < n; ++i )  
    s = s + A[ i ];
```

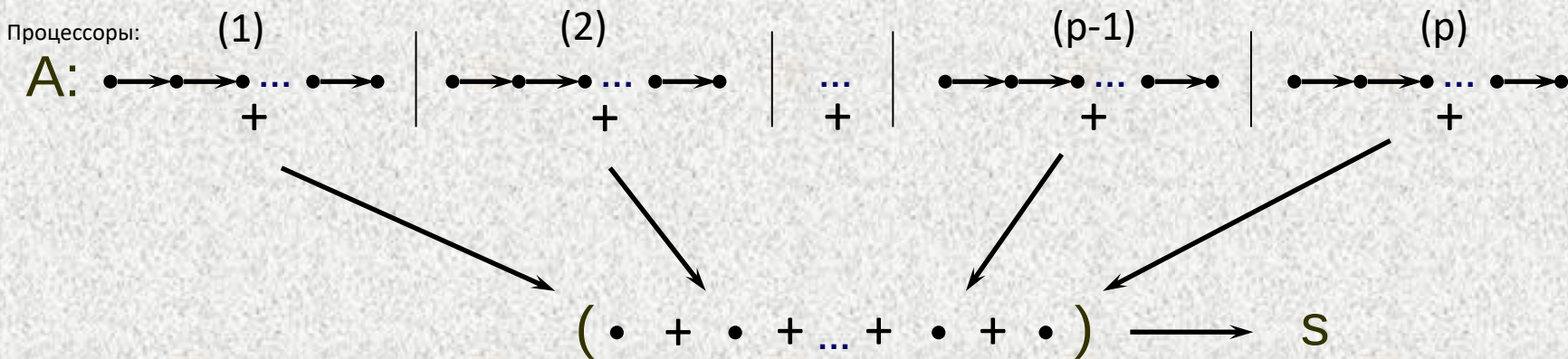


Трагедия?...

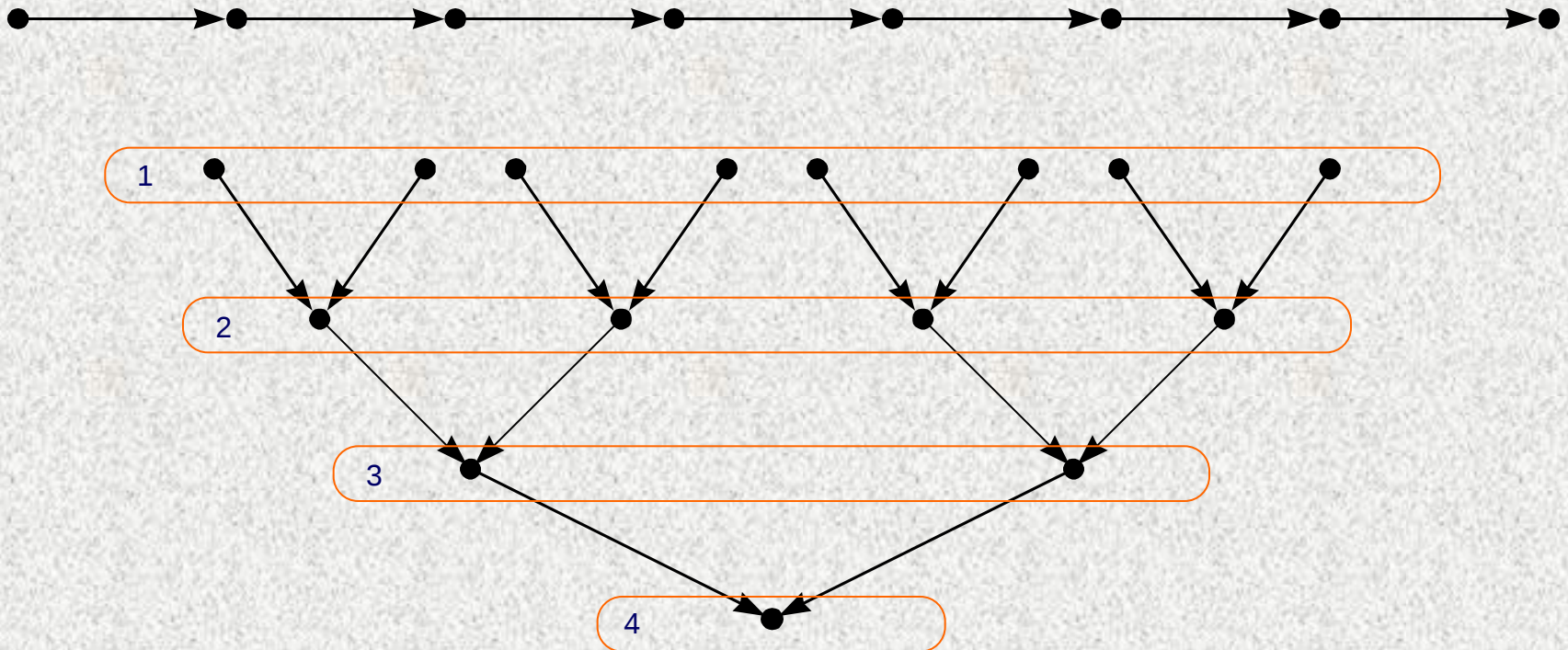
Не совсем – замена алгоритма может помочь...

Суммирование элементов массива. Замена алгоритма

```
s = 0.0;  
for ( i = 0; i < n; ++i )  
    s = s + A[ i ];
```



Суммирование элементов массива. Замена алгоритма



Простые, но фундаментальные вопросы параллельных вычислений

Как решить задачу быстрее?



Параллелизм, конвейерность...

Как решить задачу **намного быстрее?**

эффективную

миллионов

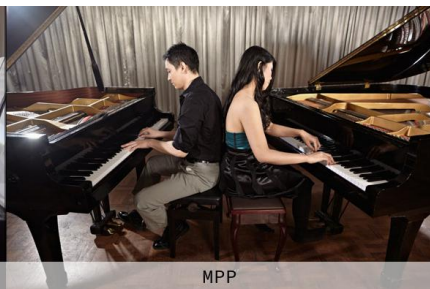
Как организовать работу нескольких человек/ядер/процессоров...?



Теория / Практика...



SMP



MPP

Компьютеры с различной архитектурой...

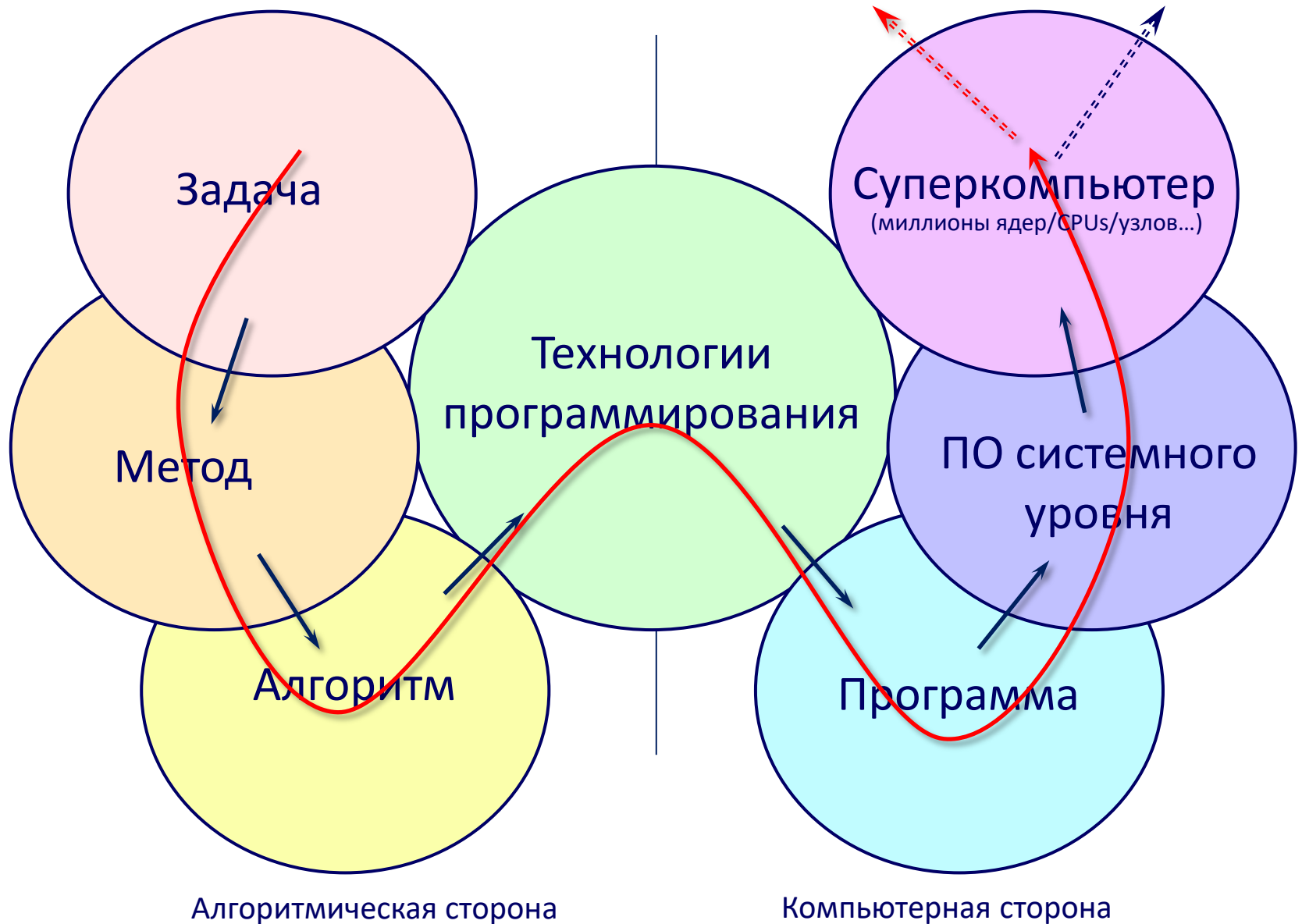


Синхронность / Асинх-ть / Модели...

Каковы основные этапы в решении задач на параллельных вычислительных системах ?

Решение задач и вычислительные системы

(Реальность) Реальная производительность $\longleftrightarrow$ Огромный разрыв! $\longleftrightarrow$ (Ожидания) Пиковая производительность



Особенность современных компьютеров – это высокая степень параллельности.

Решение задачи на параллельной вычислительной системе будет эффективным только в том случае, если на всем пути от Задачи до Компьютера нет ни одного “узкого места”!

*Центральная проблема:
отображение программ и алгоритмов на архитектуру
параллельных вычислительных систем
(co-design, кодизайн)*

Варианты сдачи зачета (экзамена) по курсу “Суперкомпьютеры и параллельная обработка данных”

Пять вариантов описаны на странице: Parallel.ru/vmk20

- Зачёт (экзамен) в форме беседы с преподавателем.
- Зачёт в форме электронного тестирования.
- Зачёт за выполнение практического задания.
- Конкурс изображений "Параллелизм вокруг нас".
- Конкурс выступлений в формате "Научный стенд-ап".



НАУЧНАЯ КОНФЕРЕНЦИЯ
Суперкомпьютерные дни в России

21-22 сентября 2020 г.

Два исключительно наполненных суперкомпьютерными событиями дня: научные секции, выставка, семинары, мастер-классы и многое другое.



Добро пожаловать!

Примите участие в конференции и получите уникальный шанс за два дня узнать все о суперкомпьютерных технологиях! Вас ждут приглашенные доклады российских и зарубежных ученых, научные секции, семинары и мастер-классы, выставка и множество других мероприятий.

*Московский государственный университет имени М.В.Ломоносова
Факультет Вычислительной математики и кибернетики*



СУПЕРКОМПЬЮТЕРЫ И ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

(часть 3)

Вл.В.Воеводин

**Зав.кафедрой Суперкомпьютеров и квантовой информатики ВМК МГУ,
Директор НИВЦ МГУ,
чл.-корр. РАН, д.ф.-м.н., профессор**

voevodin@parallel.ru

ВМК МГУ, 2020