

*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,
Директор Филиала МГУ в г.Сарове,
чл.-корр. РАН, д.ф.-м.н., профессор**

voevodin@parallel.ru

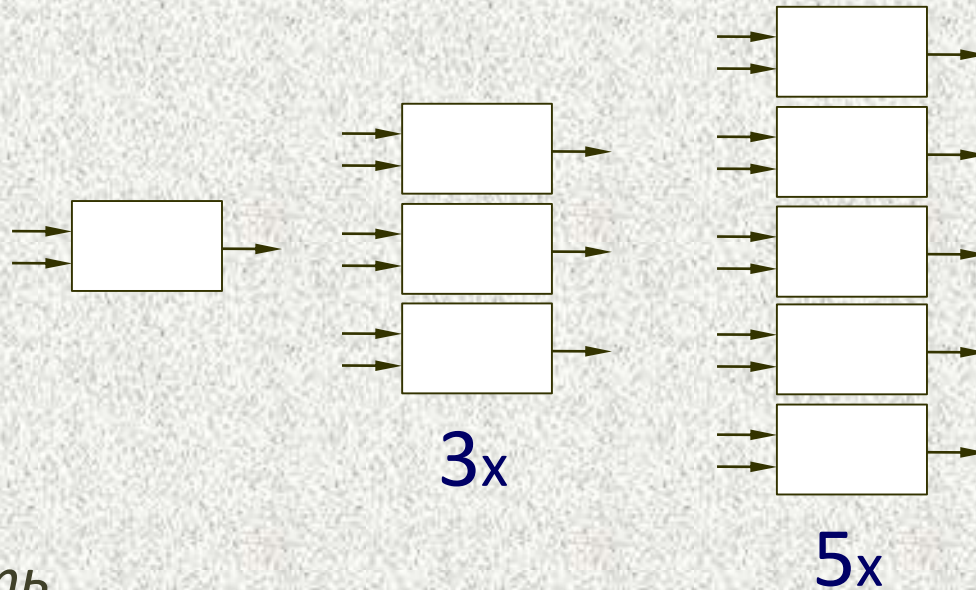
ВМК МГУ, 2024

Два вида параллельной обработки данных

- *Параллелизм*
- *Конвейерность*

Параллелизм в архитектуре компьютеров

- Параллелизм



- Конвейерность



Длина конвейера = число ступеней конвейера.

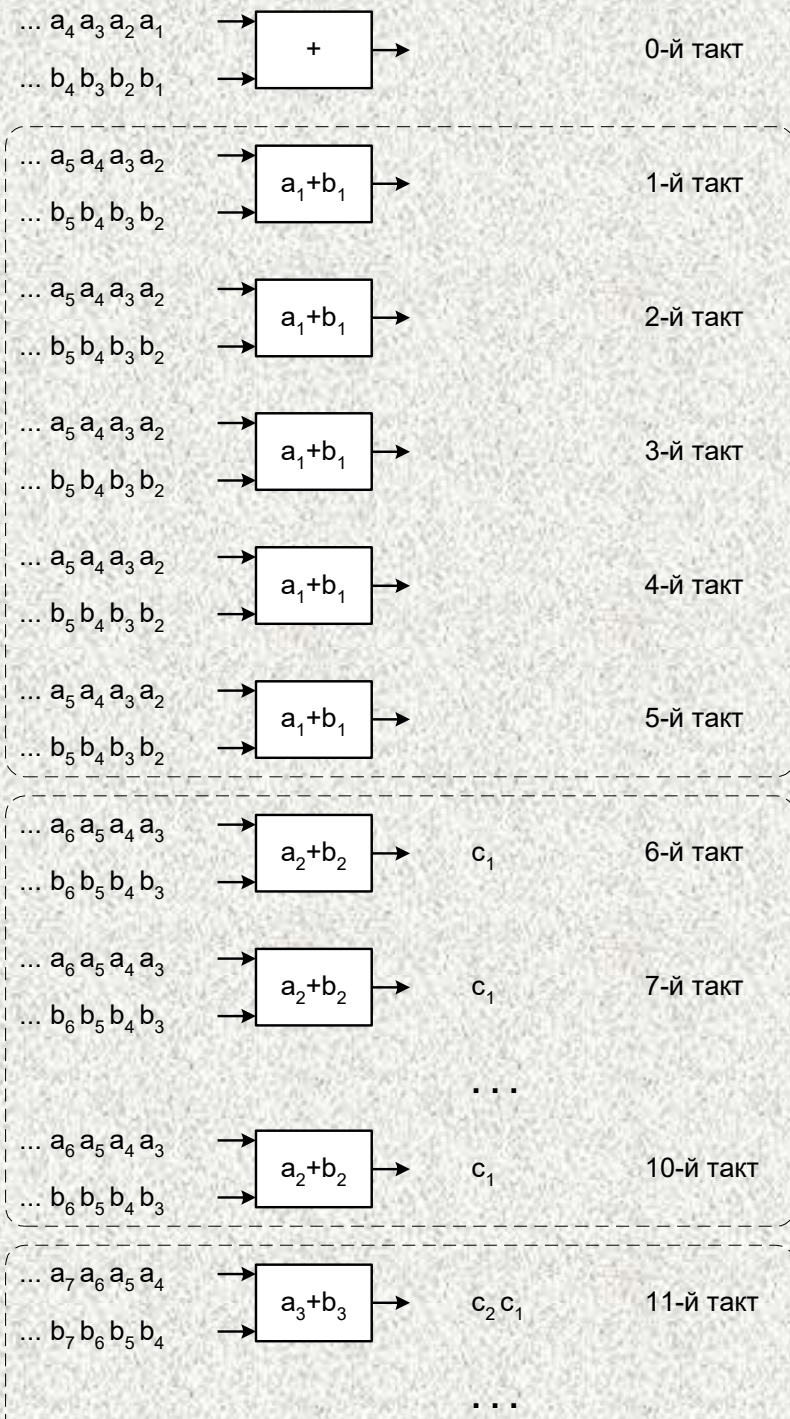
Последовательная обработка данных

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.



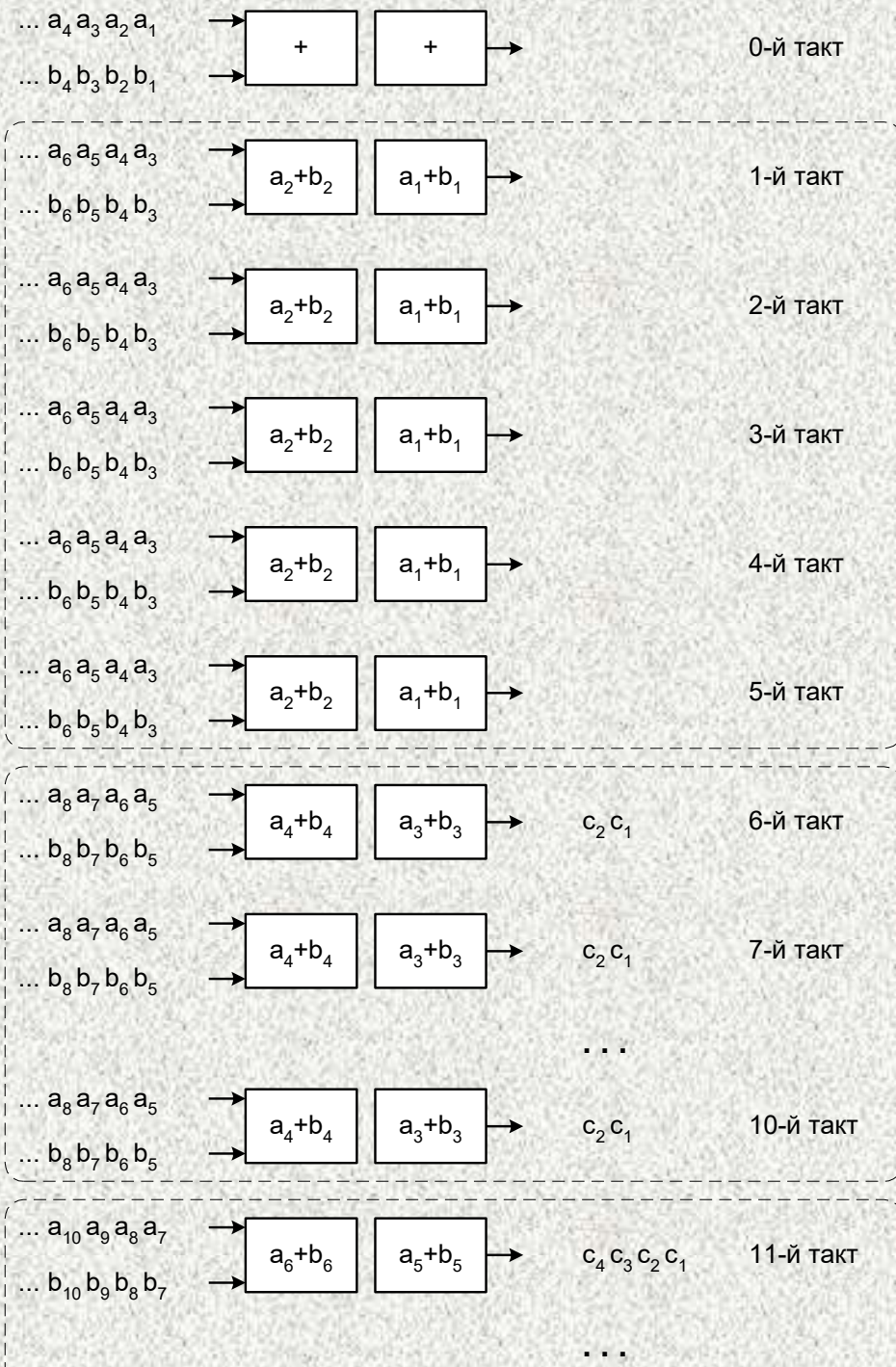
Параллельная обработка данных

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.



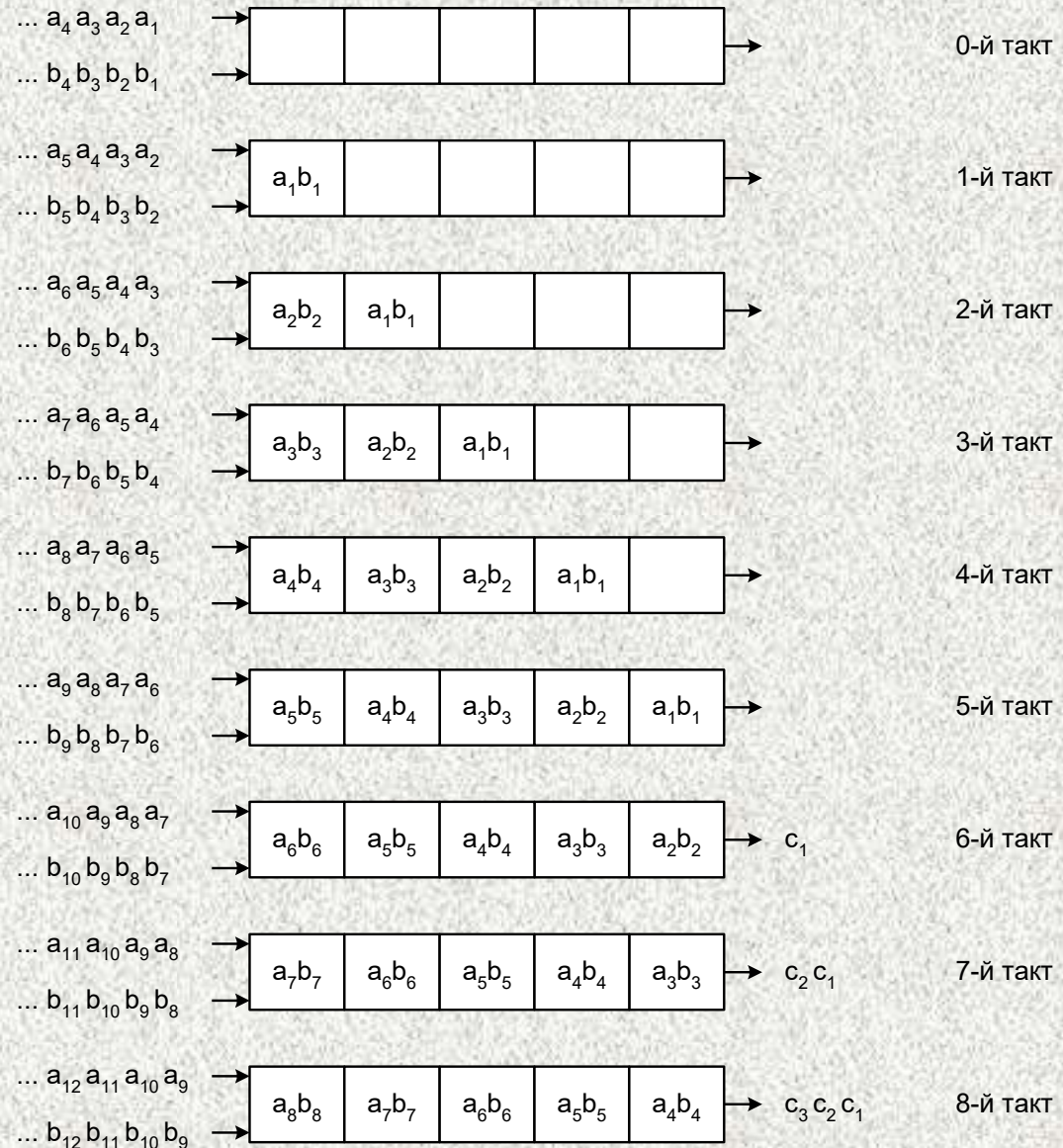
Конвейерная обработка данных

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.

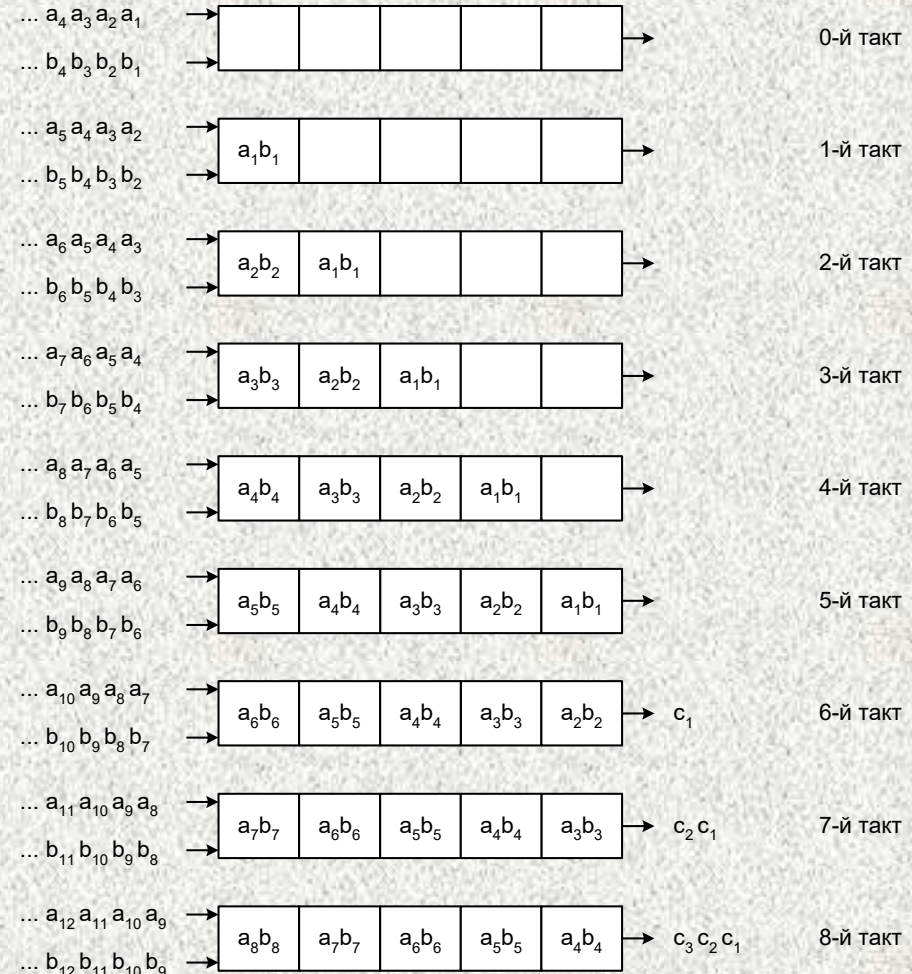
Темп выдачи результатов – каждый такт.

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...)

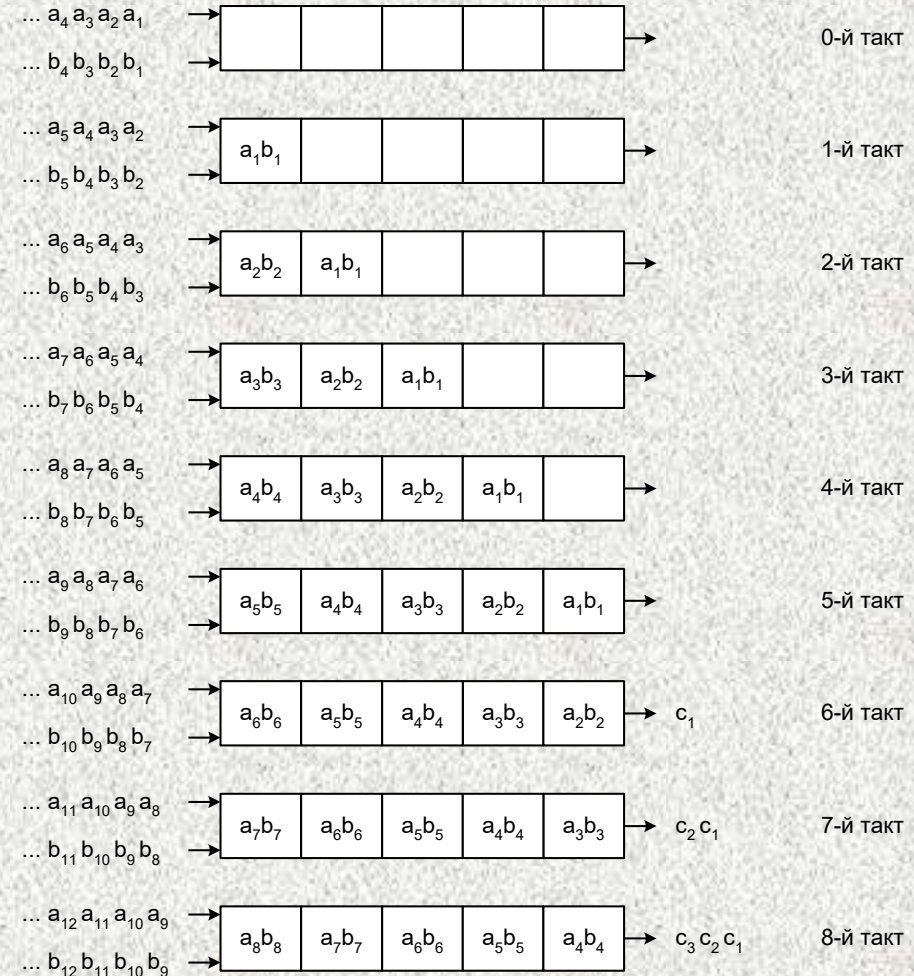


L - длина конвейера (число ступеней конвейера).

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

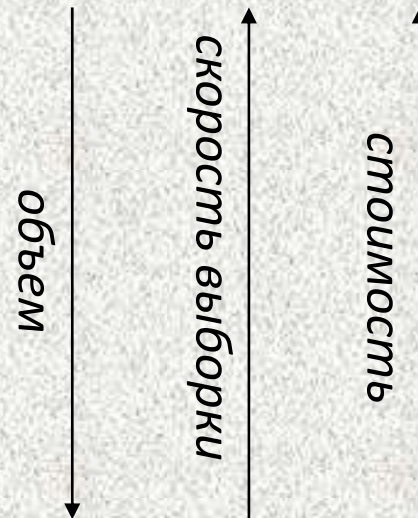
$$A_i = B_i + C_i * d, i=1..N$$

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
- ...

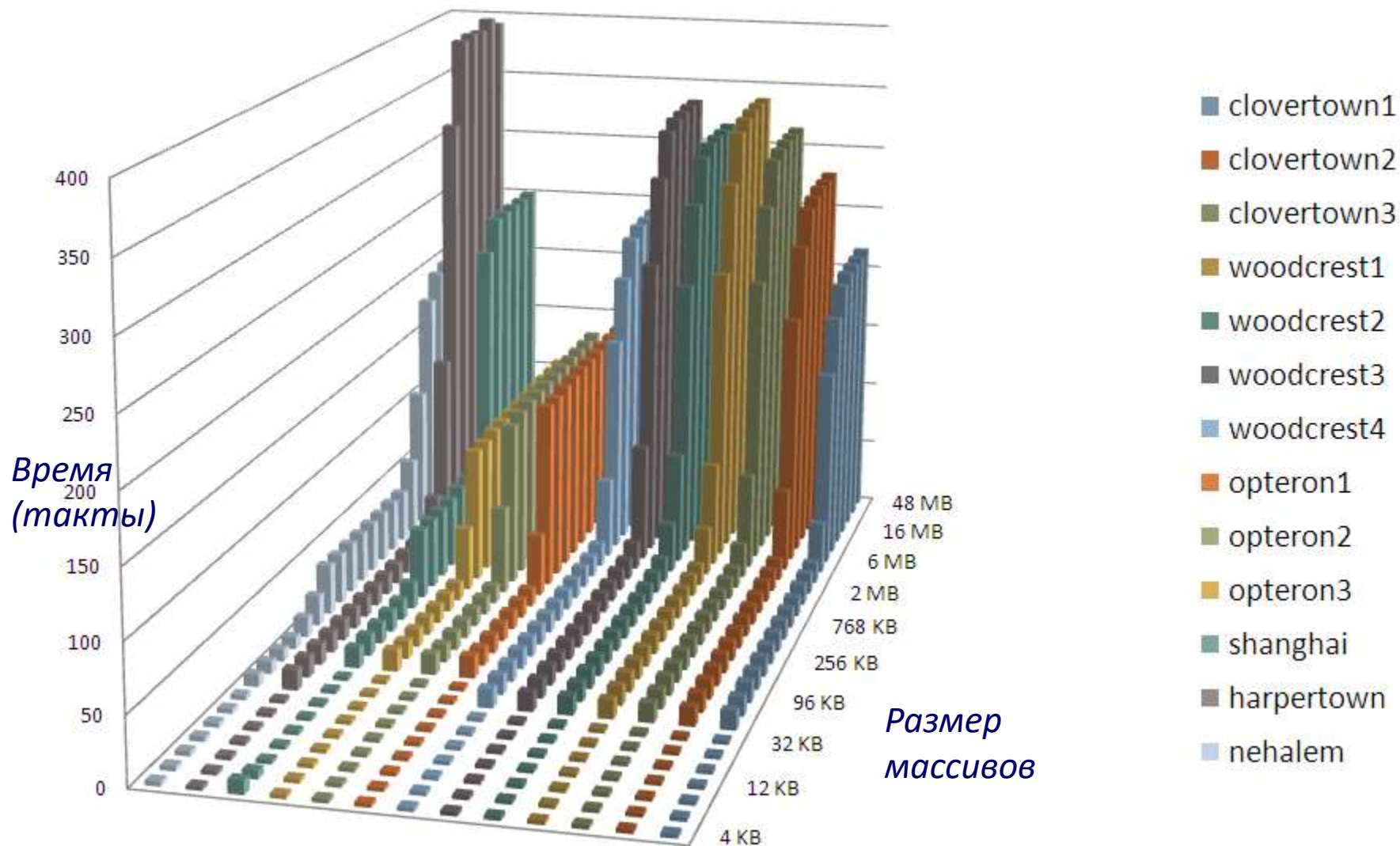


Очень важные характеристики программ:

- 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.

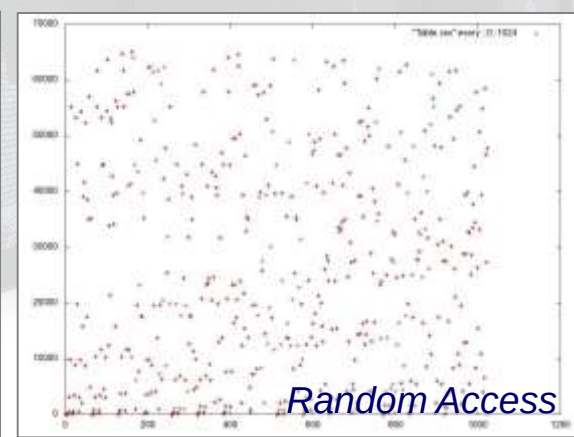
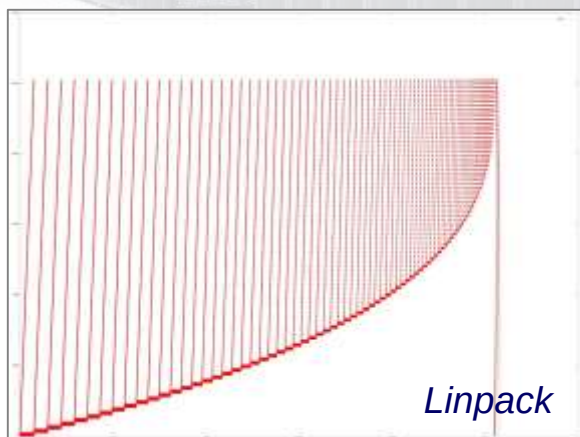
Локальность данных: множество открытых вопросов (сложности описания алгоритмов)

Как оценивать пространственную* и временную** локальность данных в программе ?

Как сравнивать пространственную и временную локальность данных разных программ ?

* Пространственная локальность данных – насколько близко друг к другу расположены последовательные обращения в память.

** Временная локальность данных – насколько часто происходят обращения к одним и тем же данным.



Профили обращения к памяти различных приложений: для каждого обращения в память показан адрес (физический или виртуальный), по которому происходит обращение.

*Современные суперкомпьютеры –
это колоссальная мощь и возможности.
Но есть ли здесь проблемы ?*

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

Предположим, что есть 100 человек и нужно найти сумму 200 чисел...

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}

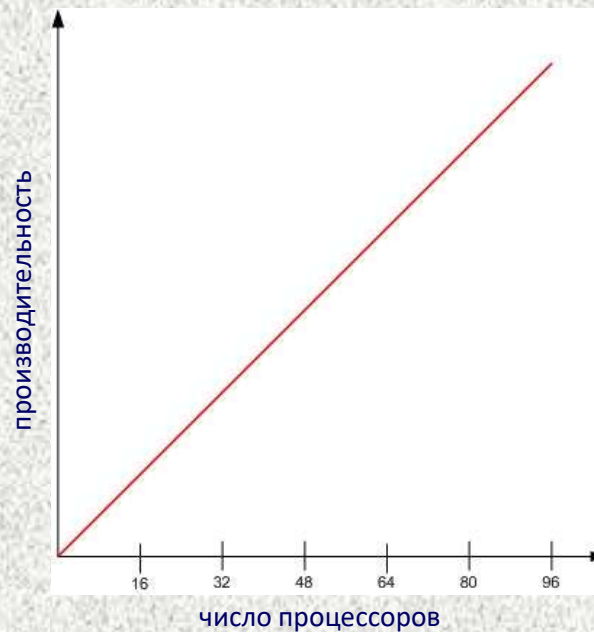
Индивидуальна для каждой программы и/или каждого компьютера. Определяется по параметрам работы программы на компьютере:

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

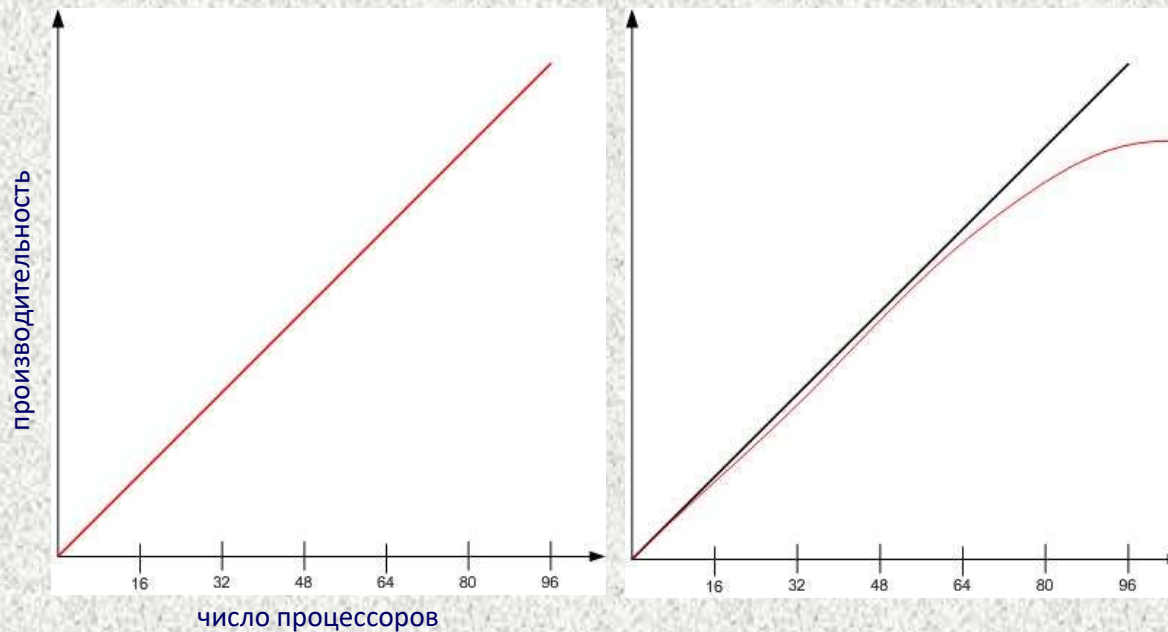
Пиковая производительность
недостижима на практике

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

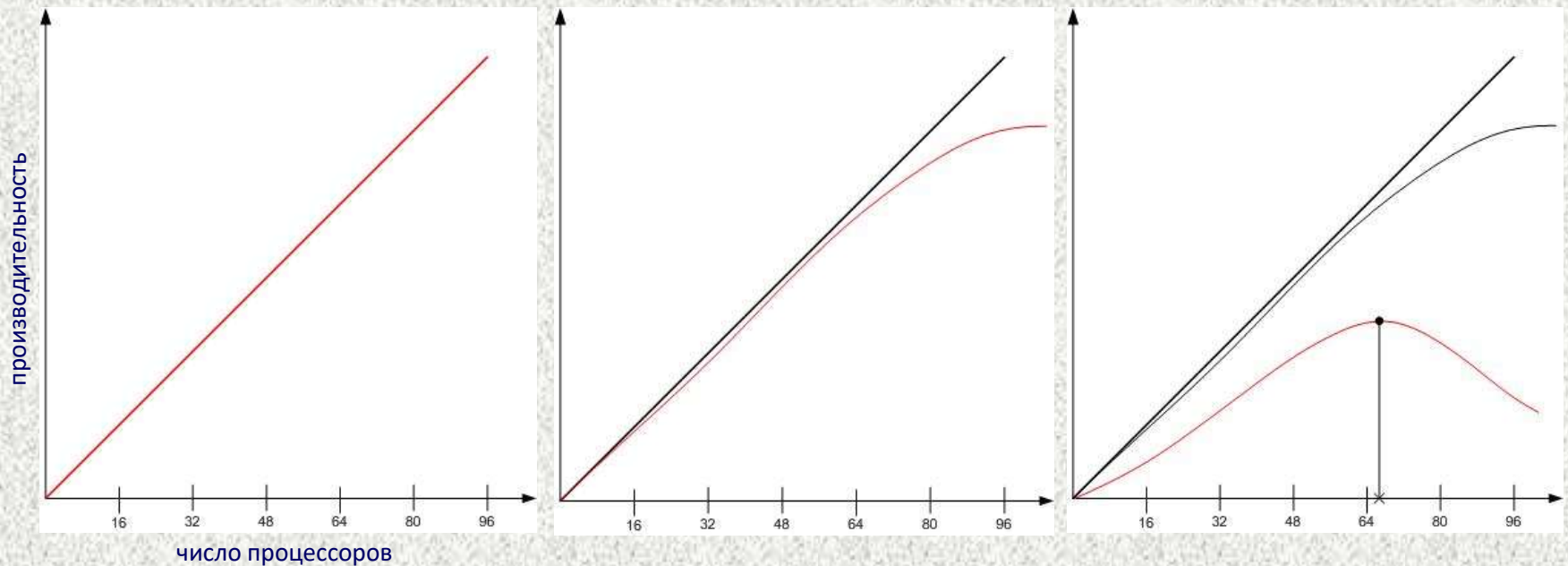
Рост производительности параллельных вычислительных систем (масштабируемость в теории)



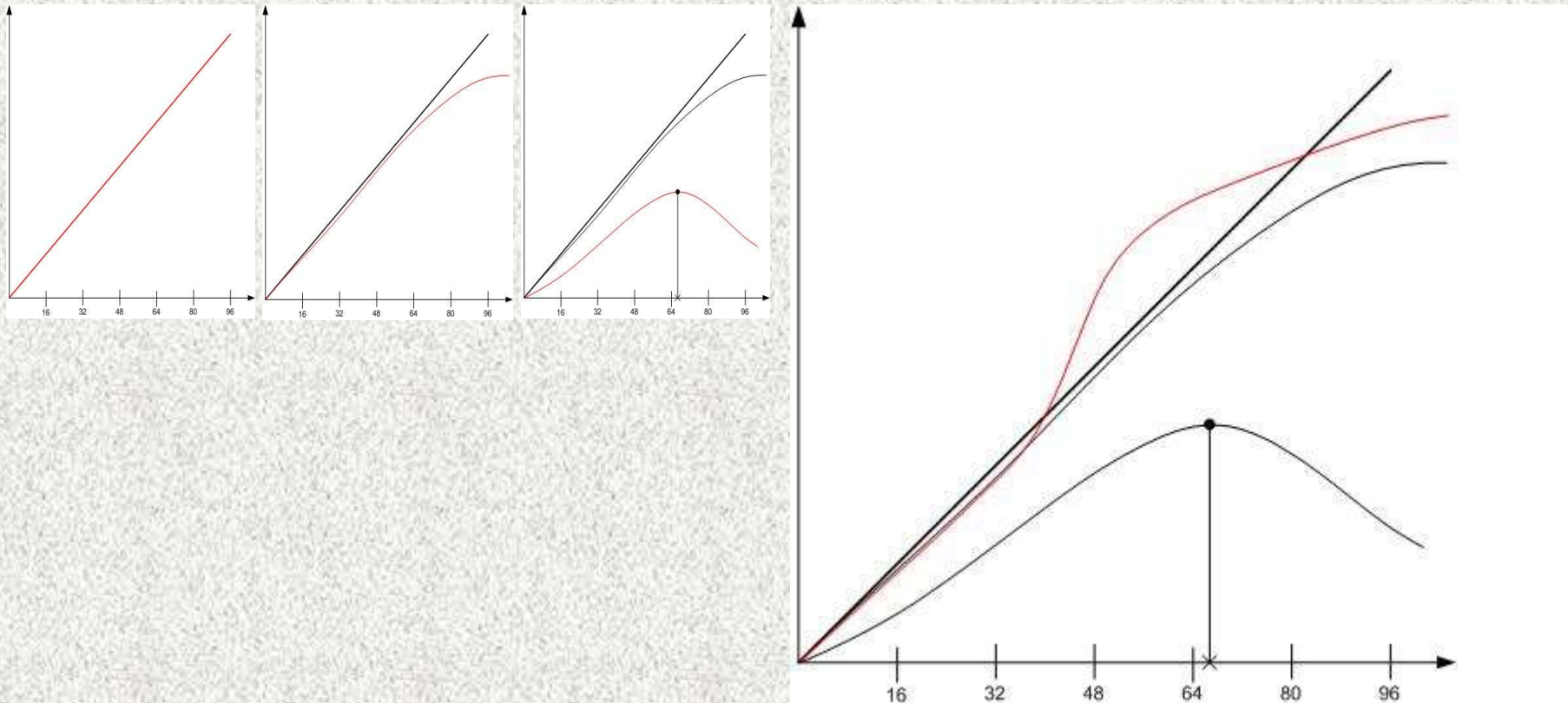
Рост производительности параллельных вычислительных систем (масштабируемость в теории и на практике)



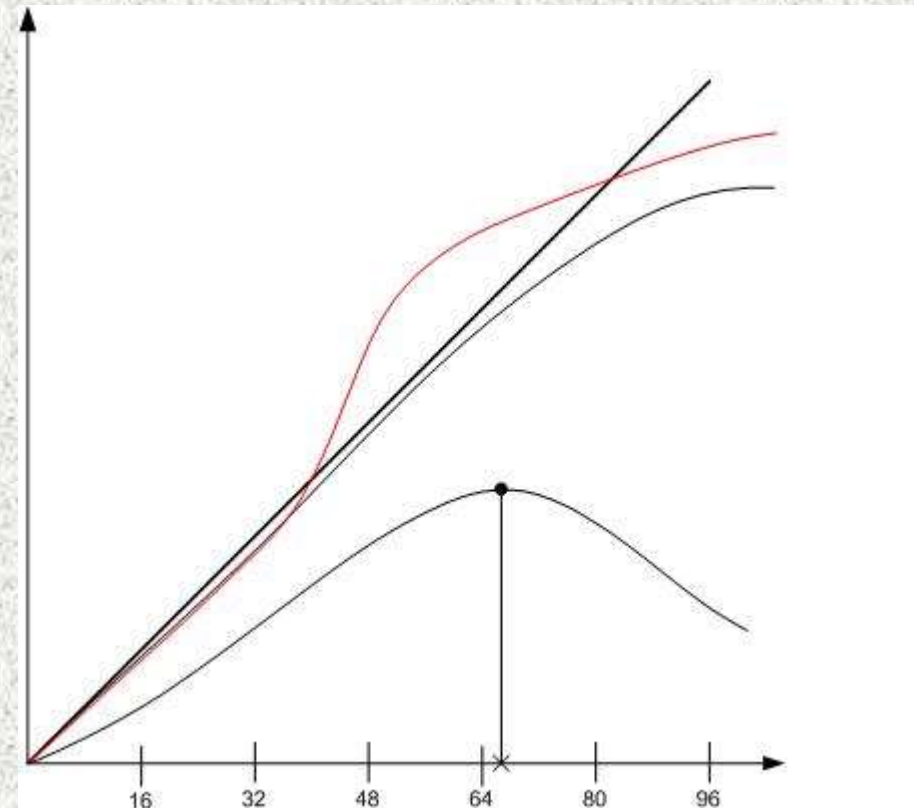
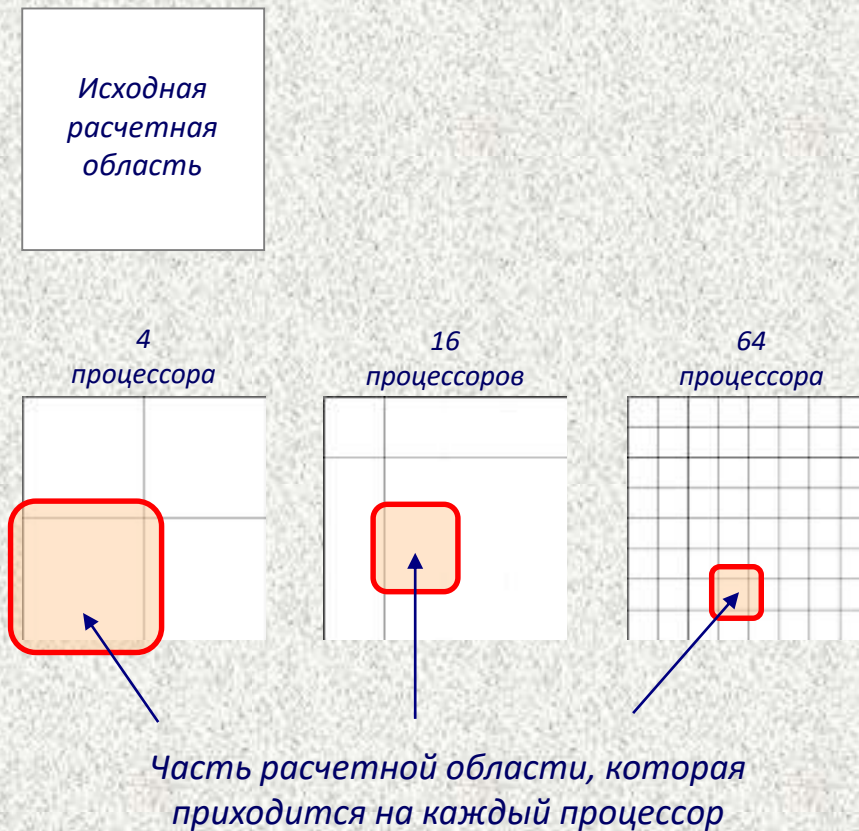
Рост производительности параллельных вычислительных систем (масштабируемость в теории и на практике)



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



Рост производительности параллельных вычислительных систем (масштабируемость 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}

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

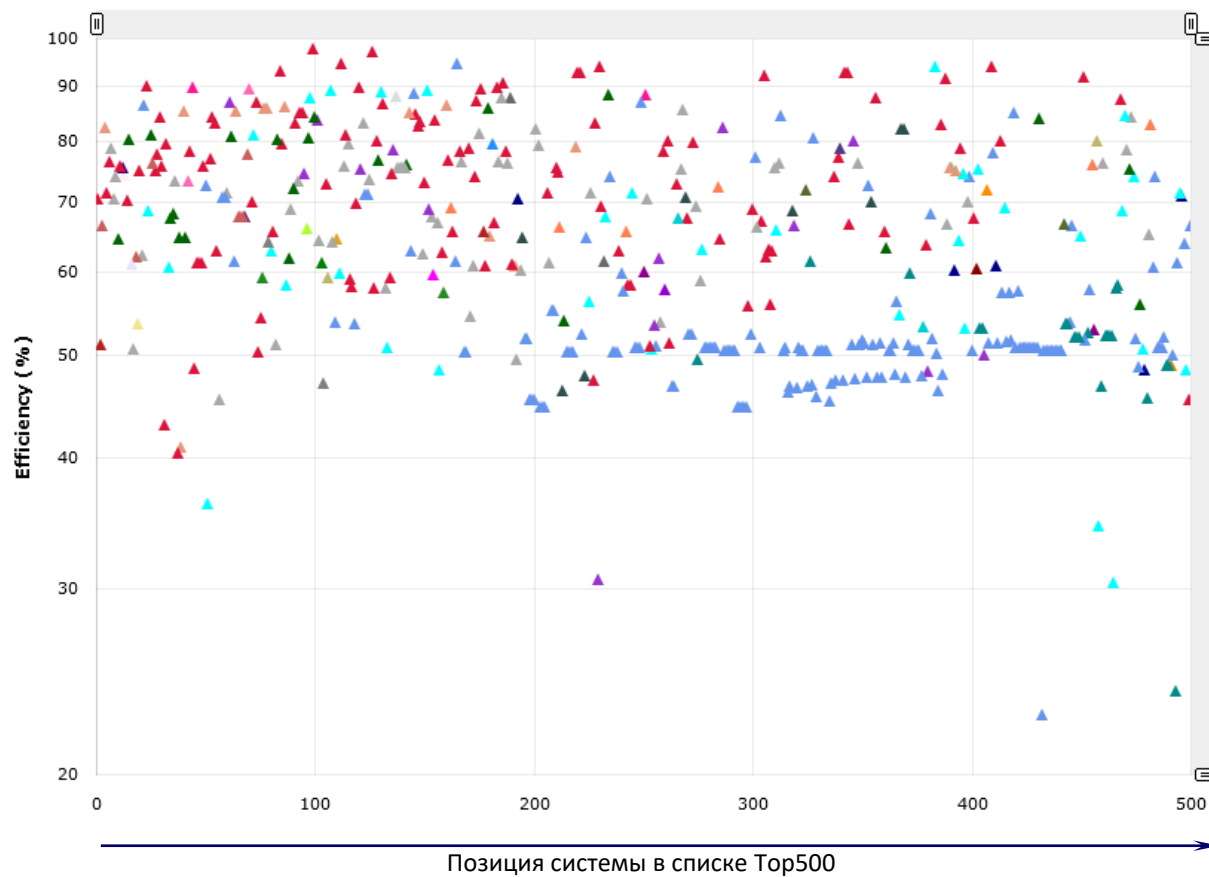
- эффективность реализации программы;
- эффективность работы компьютера (КПД);

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

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

Top500, Linpack, эффективность

(<http://top500.org>, июнь, 2024 г.)



Ключевые особенности архитектуры для обеспечения высокой производительности современных суперкомпьютеров

- Высокая степень параллелизма на всех уровнях архитектуры суперкомпьютеров (параллелизм, конвейерность).
- Обеспечение эффективной работы с иерархией памяти.

Ключевое понятие – **суперкомпьютерный кодизайн**, предполагающий обеспечение соответствия структуры и особенностей всех этапов решения задач с использованием суперкомпьютерных систем.

- Создание суперкомпьютерных систем – быстрое решение вычислительных задач и обработка высокой вычислительной нагрузки.
- Без учета особенностей архитектуры вычислительных систем высокой производительности не достичь.

Эффективность вычислительных систем (рейтинги Top500, HPCG500, Green500, Graph500)

Место в Top500	Система	Число ядер	Rmax (Pflops)	Rpeak (Pflops)	Power (kW)	Эффект-ть Top500 (%)	Место в HPCG500	Эффект-ть HPCG (%)	Место в Green500	Graph500 BFS	Graph500 SSSP
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC	8 699 904,00	1 206,00	1 714,81	22 786,00	70,33	2	0,80	13	3	-
2	Aurora - HPE Cray EX - Intel Exascale Compute Blade, Xeon	9 264 128,00	1 012,00	1 980,01	38 698,00	51,11	3	0,28	42	5	-
3	Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA	2 073 600,00	561,20	846,84		66,27	-	-	294	-	-
4	Supercomputer Fugaku - Supercomputer Fugaku, A64FX	7 630 848,00	442,01	537,21	29 899,00	82,28	1	2,91	68	1	3
5	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC	2 752 704,00	379,70	531,51	7 107,00	71,44	4	0,84	12	-	-
6	Alps - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200	1 305 600,00	270,00	353,75	5 194,00	76,33	5	1,01	14	-	-
7	Leonardo - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz,	1 824 768,00	241,20	306,31	7 494,00	78,74	6	0,99	28	-	-
8	MareNostrum 5 ACC - BullSequana XH3000, Xeon	663 040,00	175,30	249,44	4 159,00	70,28	12	0,45	15	8	-
9	Summit - IBM Power System AC922, IBM POWER9 22C	2 414 592,00	148,60	200,79	10 096,00	74,01	7	1,42	72	-	-
10	Eos NVIDIA DGX SuperPOD - NVIDIA DGX H100, Xeon Platinum	485 888,00	121,40	188,65	-	64,35	-	-	311	-	-
189	JEDI - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz,	19 584,00	4,50	5,13	67,00	87,72	94	1,15	1	-	-
128	Isambard-AI phase 1 - HPE Cray EX254n, NVIDIA Grace 72C	34 272,00	7,42	9,29	117,00	79,87	-	-	2	-	-
95	Earth Simulator - SX-Aurora TSUBASA - SX-Aurora TSUBASA	43 776,00	9,99	13,45	1 391,00	74,28	18	5,56	91	-	-
61	AOBA-S - SX-Aurora TSUBASA C401-8, Vector Engine Type 30A	64 512,00	17,22	19,82	1 389,00	86,88	13	5,49	193	-	-

Рейтинг MLPerf: расширение в 2024 году набора ML и AI тестов данного рейтинга моделями Llama 2 (70 млрд. параметров) и Stable Diffusion XL (2.6 млрд. параметров, генерация изображений).

Суммирование элементов массива. Параллелизма нет...

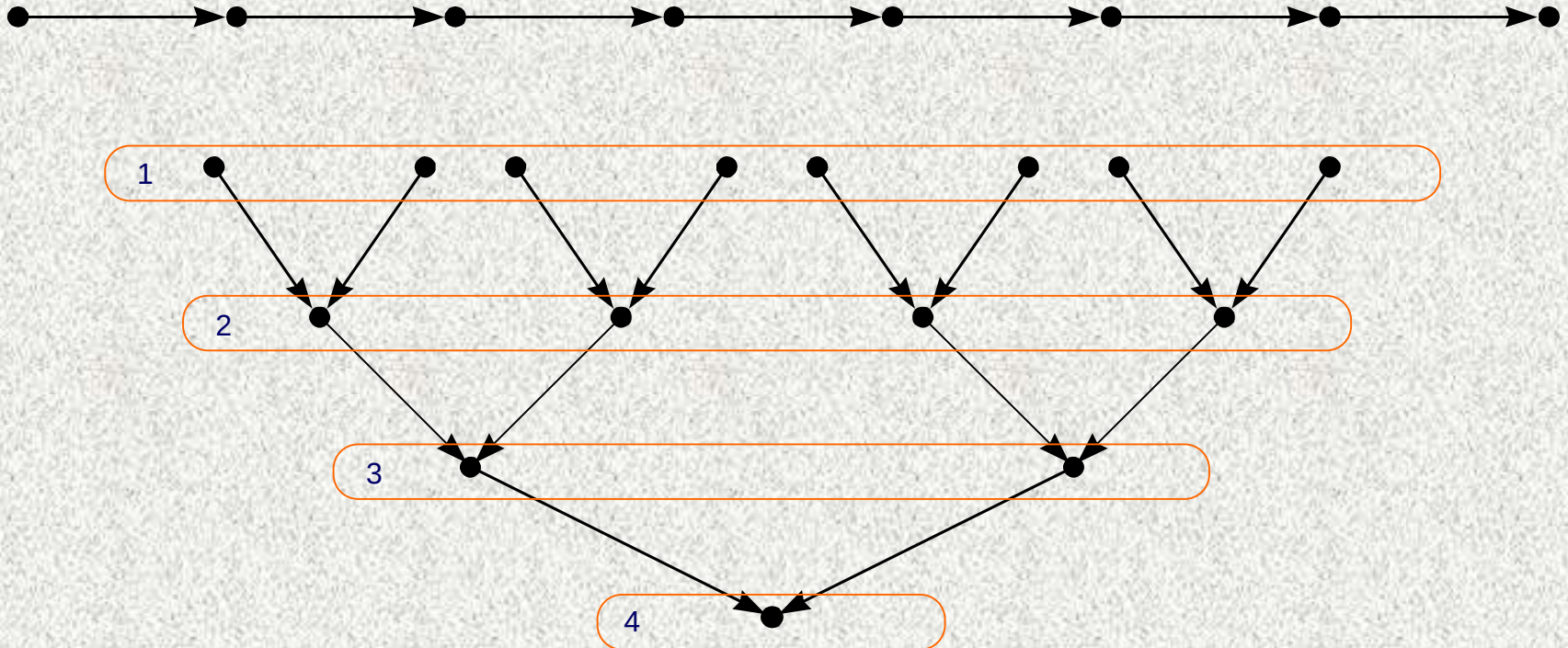
```
s = 0.0;  
for ( i = 0; i < n; ++i )  
    s = s + A[ i ];
```



Параллелизма нет – трагедия?...

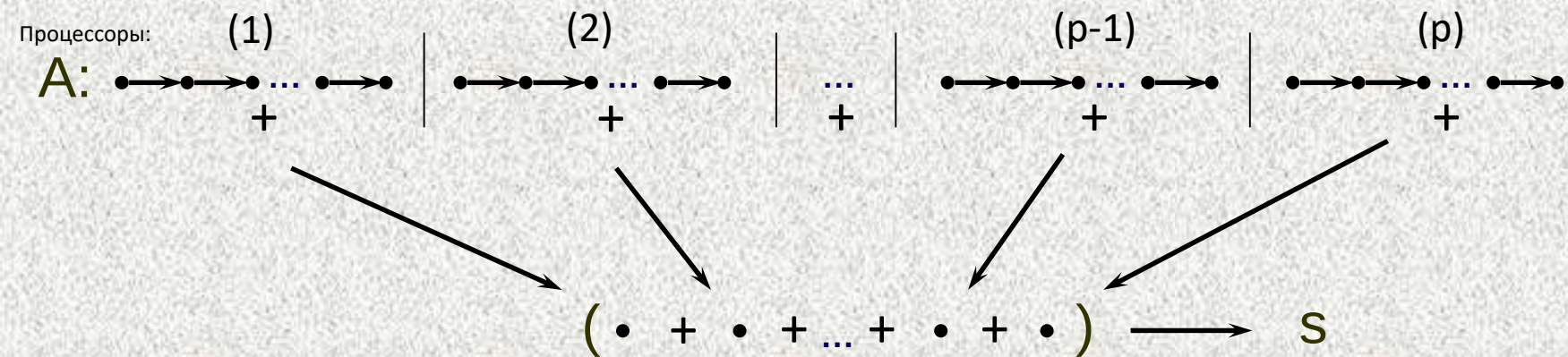
Не совсем – замена алгоритма может помочь...

Суммирование элементов массива. Замена алгоритма



Суммирование элементов массива. Замена алгоритма

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



Параллельные вычисления: важны все детали...

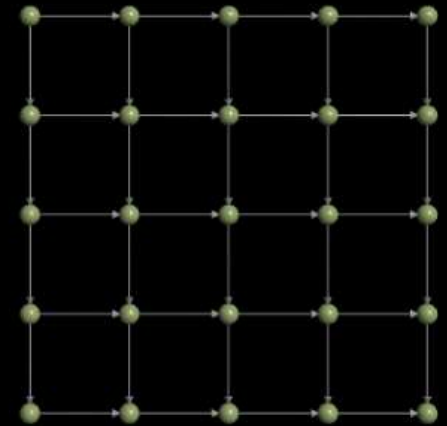
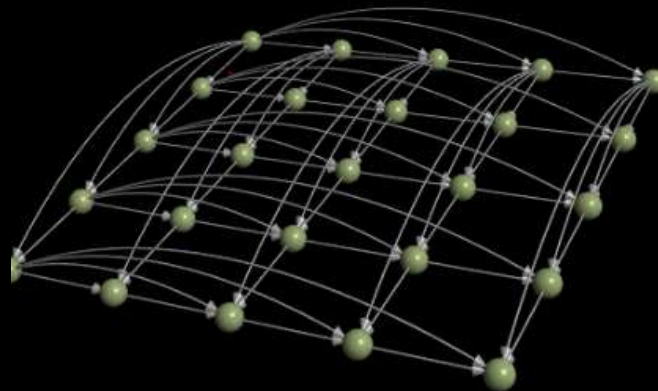
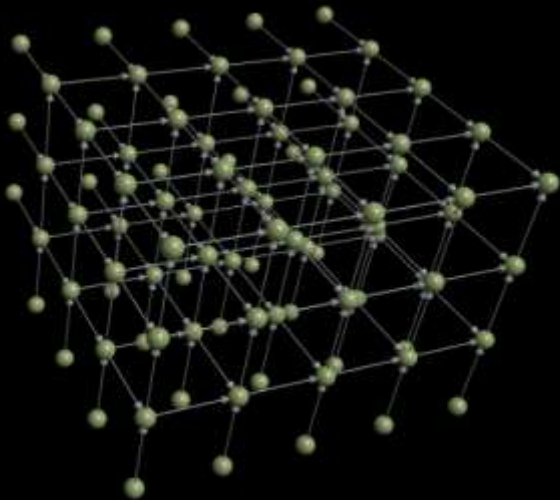
Вопрос.

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

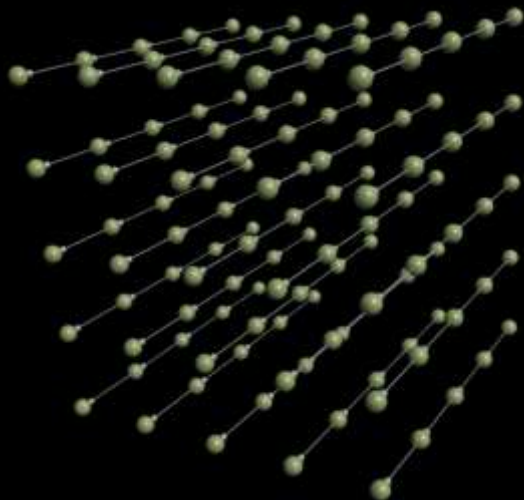
Ответ:

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

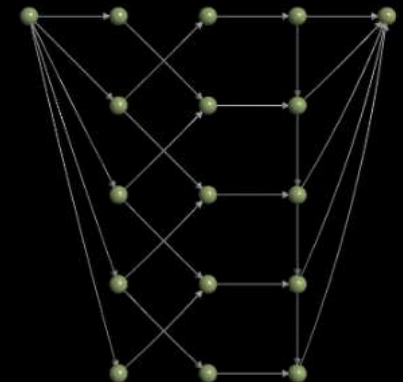
Информационная структура алгоритмов и программ



Типовые алгоритмические структуры



Огромный ресурс параллелизма



А такой структуры
быть не может ...

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...



Компьютеры с различной архитектурой...

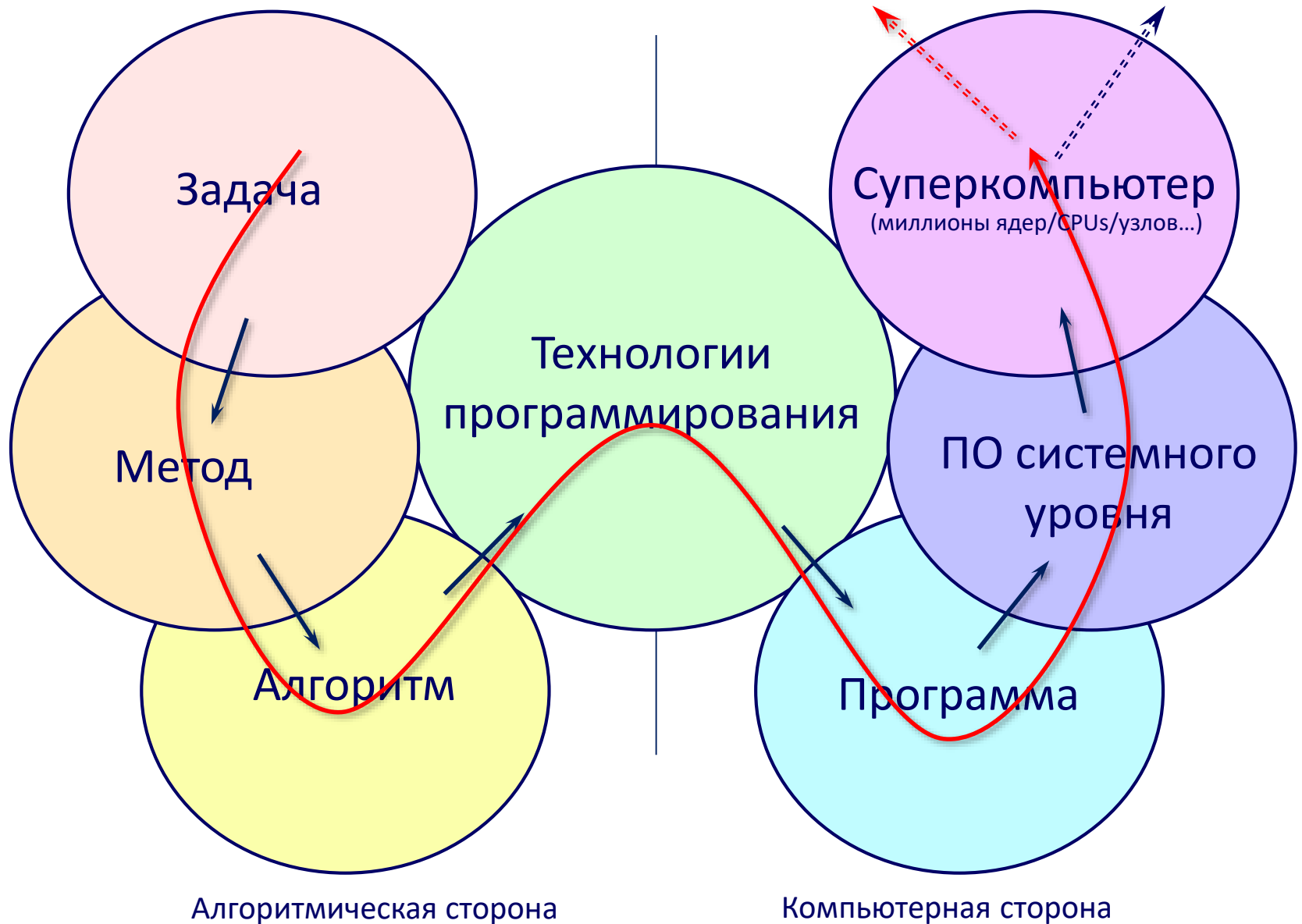


Синхронность / Асинх-ть / Модели...

Каковы основные этапы в решении задач на параллельных вычислительных системах ?

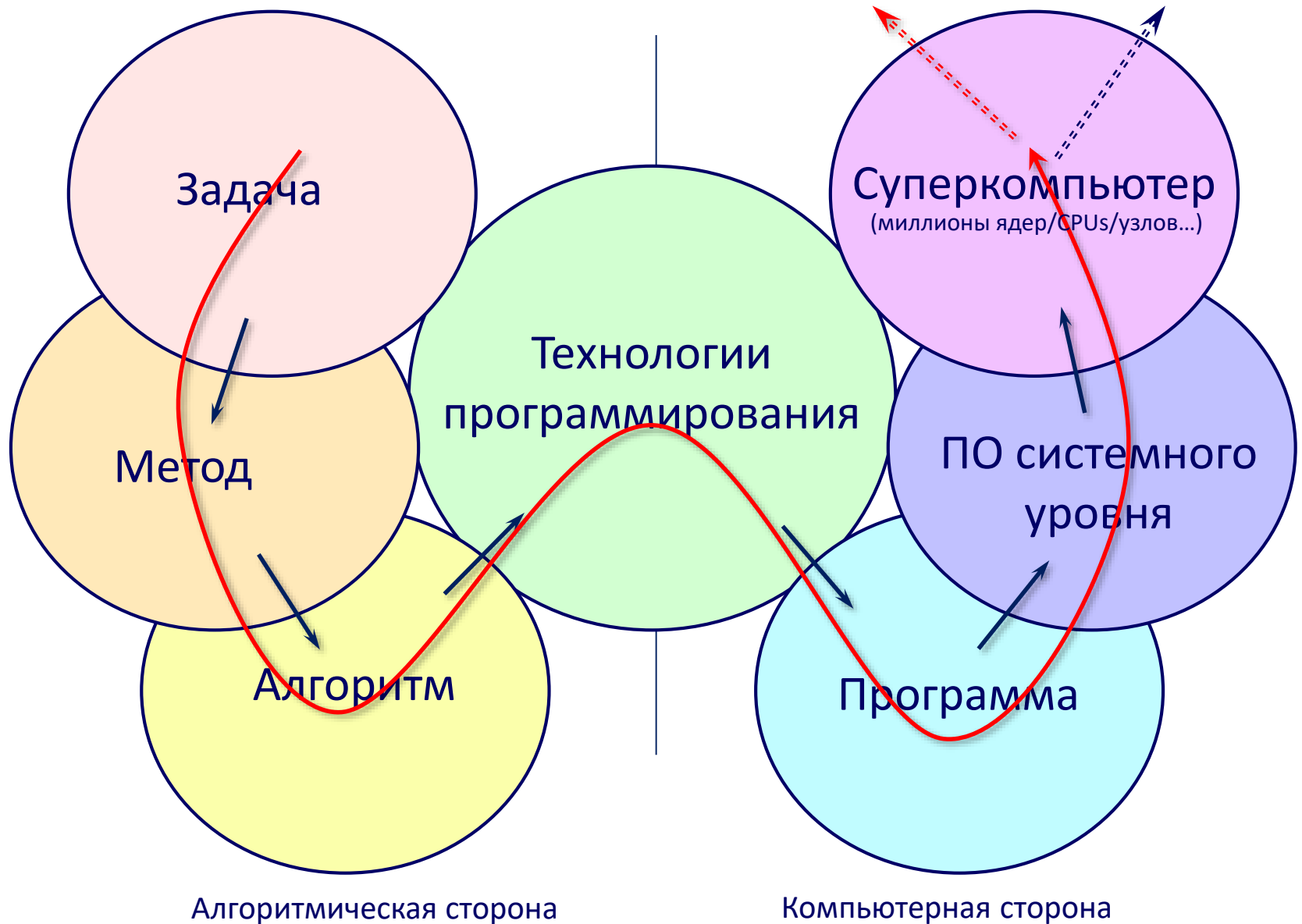
Решение задач и вычислительные системы

(Реальность) Реальная производительность ↔ Огромный разрыв! ↔ (Ожидания) Пиковая производительность



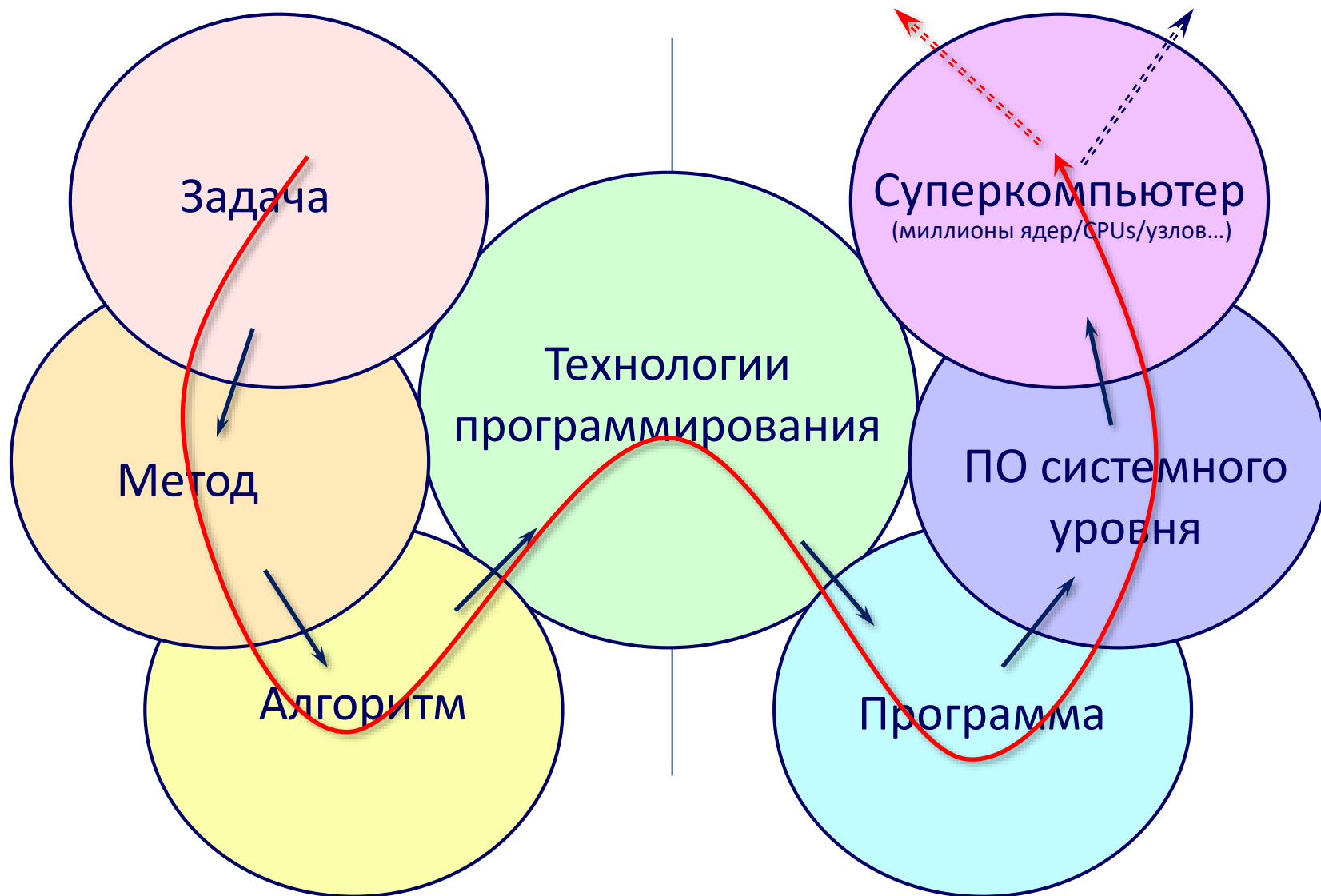
Решение задач и вычислительные системы

(Реальность) Реальная производительность ↔ Огромный разрыв! ↔ (Ожидания) Пиковая производительность



Суперкомпьютерный кодизайн

(Реальность) Реальная производительность $\xleftrightarrow{\text{Огромный разрыв!}}$ (Ожидания) Пиковая производительность

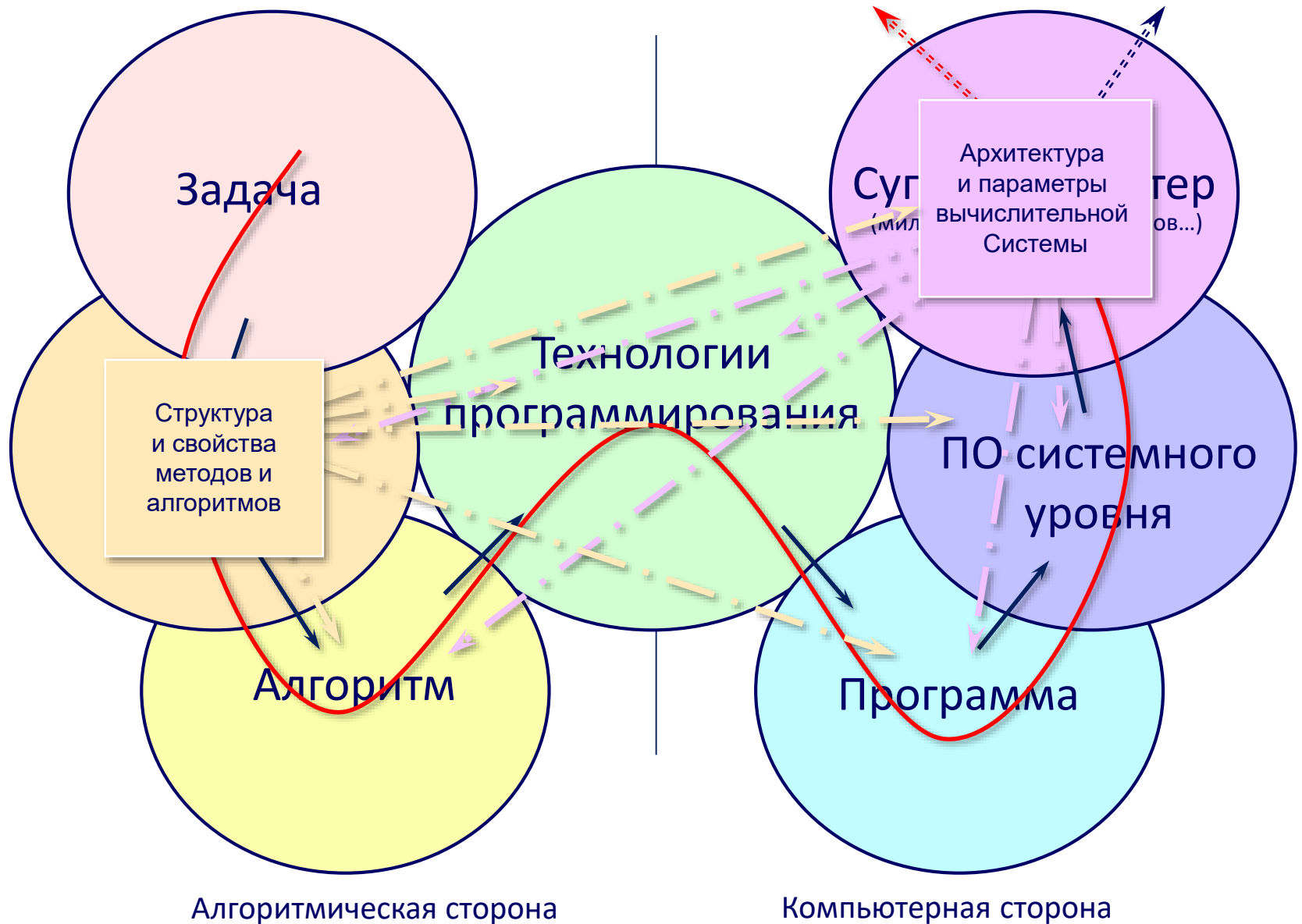


Алгоритмическая сторона

Компьютерная сторона

Суперкомпьютерный кодизайн

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



Суперкомпьютерный кодизайн: дополнительные сложности.

Масштаб суперкомпьютерных систем

(рост сложности и степени параллельности)

	Суперкомпьютер	Страна	Число ядер	Rmax (Pflops)	Rpeak (Pflops)	Power (kW)
1	Frontier - HPE Cray EX235a,	США	8,699,904	1,194.00	1,679.82	22,703
2	Fugaku, Fujitsu,	Япония	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a,	Финляндия	2,220,288	309.10	428.70	6,016
4	Leonardo - Atos BullSequana XH2000,	Италия	1,824,768	238.70	304.47	7,404
5	Summit - IBM Power System AC922,	США	2,414,592	148.60	200.79	10,096
6	Sierra - IBM Power System AC922,	США	1,572,480	94.64	125.71	7,438
7	Sunway TaihuLight - Sunway MPP,	Китай	10,649,600	93.01	125.44	15,371
8	Perlmutter - HPE Cray EX235n,	США	761,856	70.87	93.75	2,589
9	Selene - NVIDIA DGX A100,	США	555,52	63.46	79.22	2,646
10	Tianhe-2A - NUDT TH-IVB-FEP Cluster,	Китай	4,981,760	61.44	100.68	18,482

Суперкомпьютерный кодизайн: дополнительные сложности.

Неоднородность суперкомпьютерных систем

(увеличение неоднородности в мире HPC)

	Суперкомпьютер	Страна	Число ядер	Rmax (Pflops)	Rpeak (Pflops)	Power (kW)
1	Frontier - HPE Cray EX235a,	США	8,699,904	1,194.00	1,679.82	22,703
2	Fugaku, Fujitsu,	Япония	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a,	Финляндия	2,220,288	309.10	428.70	6,016
4	Leonardo - Atos BullSequana XN2000,	Италия	1,824,768	238.70	304.47	7,404
5	Summit - IBM Power System AC922,	США	2,414,592	148.60	200.79	10,096
6	Sierra - IBM Power System AC922,	США	1,572,480	94.64	125.71	7,438
7	Sunway TaihuLight - Sunway MPP,	Китай	10,649,600	93.01	125.44	15,371
8	Perlmutter - HPE Cray EX235n,	США	761,856	70.87	93.75	2,589
9	Selene - NVIDIA DGX A100,	США	555,52	63.46	79.22	2,646
10	Tianhe-2A - NUDT TH-IVB-FEP Cluster,	Китай	4,981,760	61.44	100.68	18,482

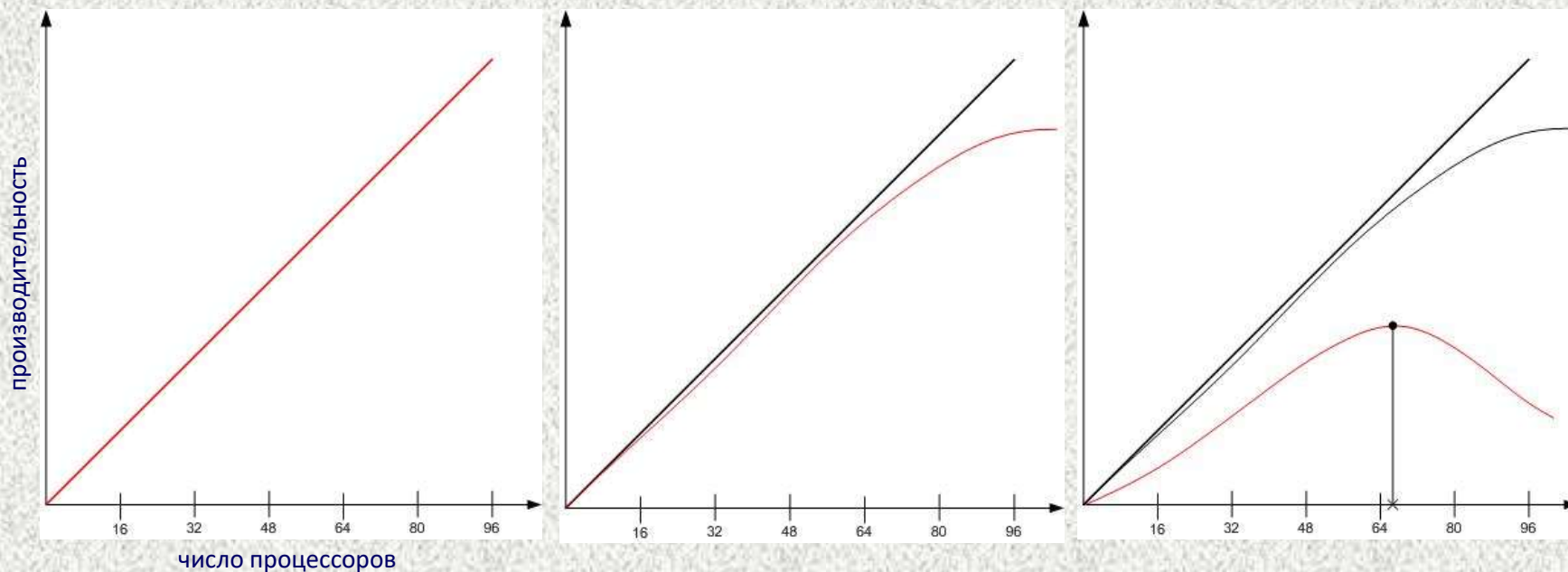
9 систем из первых 10 систем списка Top500 – с неоднородной архитектурой.
Основной вопрос: как использовать неоднородные системы?

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)*

Масштабируемость параллельных программ и параллельных вычислительных систем (влияют все этапы!)



Возможные причины снижения масштабируемости:

- Многократное порождение/уничтожение нитей.
- Излишние барьеры для синхронизации работы нитей.
- Неравномерное распределение вычислительной нагрузки.
- Множественные конфликты по адресатам при передаче данных.
- Конечный параллелизм.
- Шум в операционной системе.
- Плохая организация областей вычислений и коммуникаций.
- Излишне мелкая гранулярность при передаче данных.
- Отсутствие привязки нитей к ядрам.
- и т.д., и т.п. ...

Суперкомпьютерный кодизайн – что важно?

*Производительность? Эффективность? Загруженность?
Масштабируемость?*

Время решения задачи !



Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers

David H. Bailey

Technical Report RNR-91-020, 1991

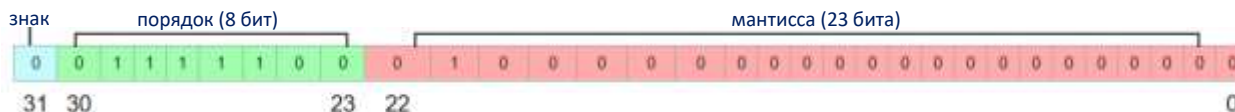
Abstract

Many of us in the field of highly parallel scientific computing recognize that it is often quite difficult to match the run time performance of the best conventional supercomputers. This humorous article outlines twelve ways commonly used in scientific papers and presentations to artificially boost performance rates and to present these results in the "best possible light" compared to other systems.

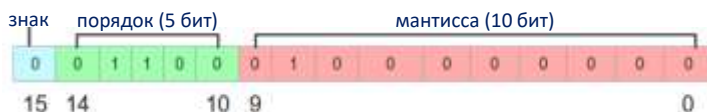
1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this fact.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on Crays.
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If MFLOPS rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
9. Quote performance in terms of processor utilization, parallel speedups or MFLOPS per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

Форматы представления вещественных чисел (кодизайн: производительность, компактность)

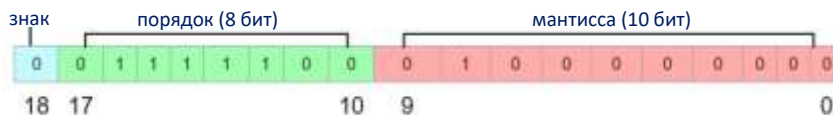
IEEE 754 single-precision 32-bit float



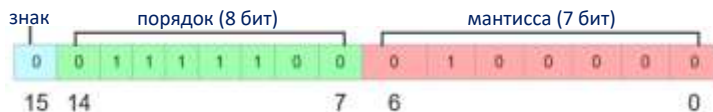
IEEE half-precision 16-bit float



NVIDIA's Tensor float (19 бит)



bfloat16



8-bit float (1.4.3 minifloat)



Форматы и разрядность современных компьютеров:

- вещественные числа,
- целые числа,
- 128/64/32/24/19/16/8/4 бит

Суперкомпьютер Fugaku:
Int: 8,16,32,64 бум;
Float: 16,32,64 бум

19 августа 2024

Самый мощный чип в мире: Nvidia представила флагманский ИИ-ускоритель Blackwell B200

Казалось бы, совсем недавно на рынок вышел сверхмощный ИИ-чип Nvidia H100, однако Nvidia и не показывает намерения останавливаться. Новые ускорители, Blackwell B200 и GB200, поднимают энергоэффективность и производительность ИИ-систем на новый уровень.

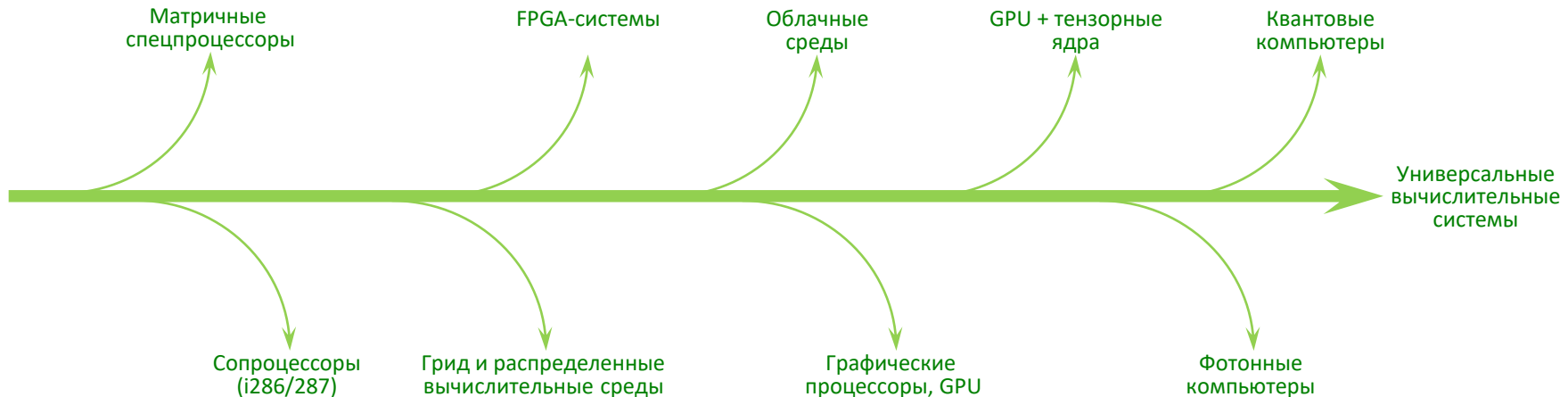
Кристалл Blackwell B200 содержит 208 миллиардов транзисторов и обеспечивает до 20 петафлопс вычислительной мощности (FP4).

	Hopper	Ampere	Turing	Volta
Операции, поддерживаемые тензорными ядрами	FP64, TF32, bfloat16, FP16, FP8, INT8	FP64, TF32, bfloat16, FP16, INT8, INT4, INT1	FP16, INT8, INT4, INT1	FP16
Операции, поддерживаемые ядрами CUDA	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, INT8	FP64, FP32, FP16, INT8

Развитие архитектур вычислительных систем для эффективного решения классов задач (кодизайн и специализация)

Возможные причины появления ВС специальной архитектуры:

- Эффективность
- Производительность
- Цена/производительность
- Энергоэффективность
- ...



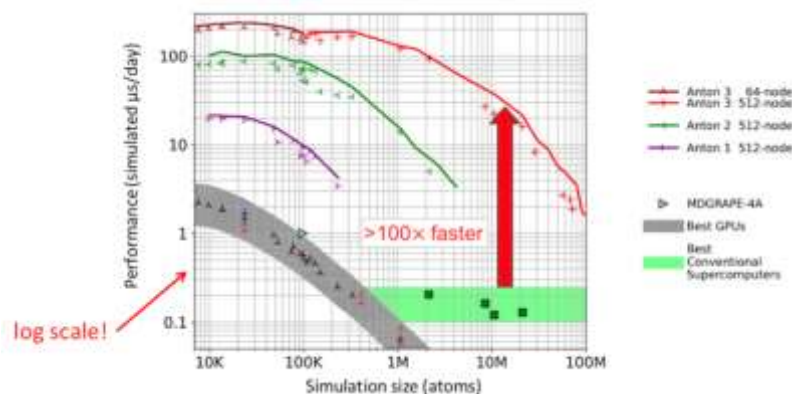
Ландшафт доступных компьютерных архитектур

(Насколько разнообразен компьютерный мир сегодня?)



*И мы, безусловно, должны думать о правильном подборе архитектуры...
А меняются ли алгоритмы при изменении архитектуры?...*

Суперкомпьютер Anton – апофеоз кодизайна для моделирования молекулярной динамики (D.E. Shaw Research company)

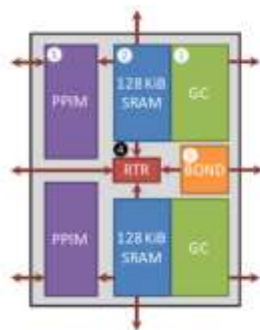


Молекулярная динамика на разных платформах.



	ANTON	ANTON 2	ANTON 3
Tape-out	2007	2012	2020
CPU cores	8+4+1	66	528*
PPIMs	32	76	528*
Flex SRAM	0.125 MiB	4 MiB	66 MiB*
Atoms / node	460	8,000	110,000*
Clock frequency	0.485/0.970 GHz	1.65 GHz	2.8+ GHz
Channel bandwidth	0.607 Tbps	2.7 Tbps	5.6+ Tbps
Process node	90 nm	40 nm	7 nm
Transistors	0.2 G	2.0 G	31.8 G
Die size	299 mm ²	410 mm ²	451 mm ²
Power	30 W	190 W	360 W

Развитие процессоров семейства Anton.



Evolutionary changes

- Support additional functional forms
- Increase memory capacity
- Tune instruction set for MD application
- Increase code density

Revolutionary changes

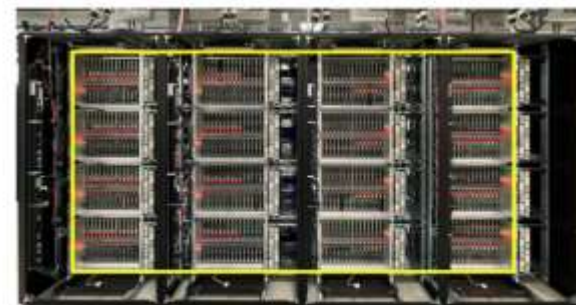
- Co-locate compute resources
- Specialize bonded force computation
- Double effective density of pairwise interaction calculation
- Implement fine-grained synchronization within memory and network



8x8 nodes



2x64 nodes



512 nodes

Суперкомпьютер Anton – апофеоз кодизайна для моделирования молекулярной динамики (D.E. Shaw Research company)

Hot Chips Conference 2021...

Two interesting questions came from Jeffrey Vetter at Oak Ridge National Labs, who asked if the system could scale beyond 512 nodes for larger MD simulations, and whether other types of molecular dynamics applications were being considered, such as those focused on material science.

Adam Batson, D.E. Shaw Research, said, “I can tell you that the hardware is physically capable of scaling larger than 512 nodes in terms of the network in the link layer. But the machine is definitely designed to operate normally for where a single molecular dynamic simulation runs on at most 512 nodes...

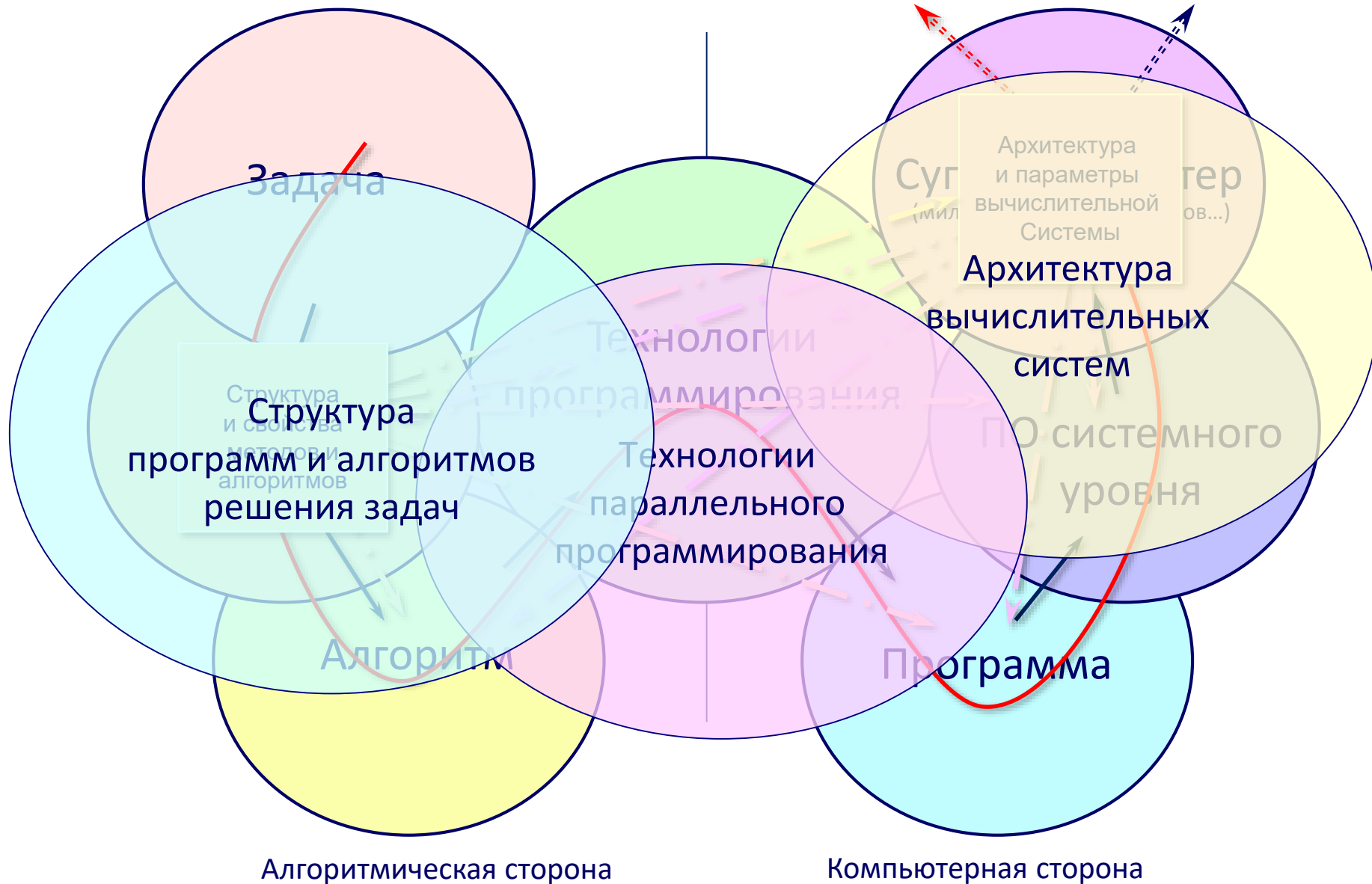
“For the other applications, in particular material science, it’s possible there are applications that would benefit from Anton. We’ve looked at this somewhat within D.E. Shaw Research, but not a whole lot. Like I said before, you know, we’re not a computer company, our focus is on curing diseases and easing human misery and we just don’t spend a lot of time working on things outside of that scope.”

Персональный кодизайн (структура курса)

(Реальность)
Реальная
производительность

Огромный
разрыв !

(Ожидания)
Пиковая
производительность



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

Пять вариантов описаны на странице: Parallel.ru/vmk24

- Экзамен в форме беседы с преподавателем.
- Электронное тестирование.
- Выполнение практического задания.
- Конкурс изображений "Параллелизм вокруг нас".
- Конкурс выступлений в формате "Научный стенд-ап".

Суперкомпьютерные дни в России 2024

международная научная конференция

23 - 24 сентября

два дня — множество параллельных событий

*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,
Директор Филиала МГУ в г.Сарове,
чл.-корр. РАН, д.ф.-м.н., профессор**

voevodin@parallel.ru

ВМК МГУ, 2024