

*Суперкомпьютерные технологии, параллельные вычисления  
и структура алгоритмов*

*Вл.В.Воеводин*

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

*voevodin@parallel.ru*

## *А.П.Ершов – первое знакомство...*



Ершов А.П. Введение в теоретическое программирование. Беседы о методе.  
Издательство "Наука", 1977



# Суперкомпьютерный комплекс Московского университета



# Вычислительная аэродинамика мельчайших насекомых

Полилов А.А.<sup>1</sup>, Фарисенков С.Э.<sup>1</sup>, Лапина Н.А.<sup>1</sup>, Коломенский Д.С.<sup>2</sup>

1) МГУ имени М.В.Ломоносова

2) Сколковский Институт науки и технологий

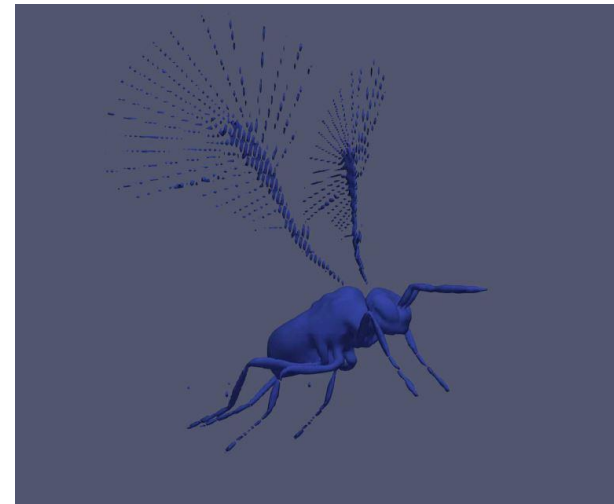
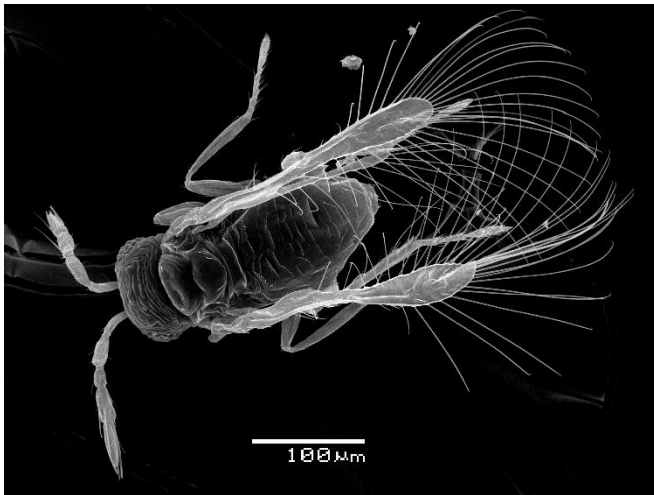
**ЗАДАЧА.** Расчёт аэродинамических сил, действующих в полёте на миниатюрных насекомых с длиной тела менее 0,5 мм.

**СТРАТЕГИЯ.** Численное решение уравнений Навье-Стокса методом конечных разностей с динамической адаптацией сетки и использованием трехмерных гоометрических моделей полученных на основе микрофотографий со сканирующего электронного микроскопа.

**ЦЕЛЬ.** Анализ аэродинамических свойств перистых крыльев мельчайших насекомых.

**ЭФФЕКТ.** Объяснение механизмов полета мельчайших насекомых.

**ИСПОЛЬЗОВАНИЕ.** Энтомология, Зоология, Эволюция, Биомеханика, Аэрогидромеханика.



# Многомасштабное моделирование нанодисперсных полимерных и супрамолекулярных систем

Комаров П.В., Халатур П.Г., Малышев М.Д.  
МГУ имени М.В.Ломоносова

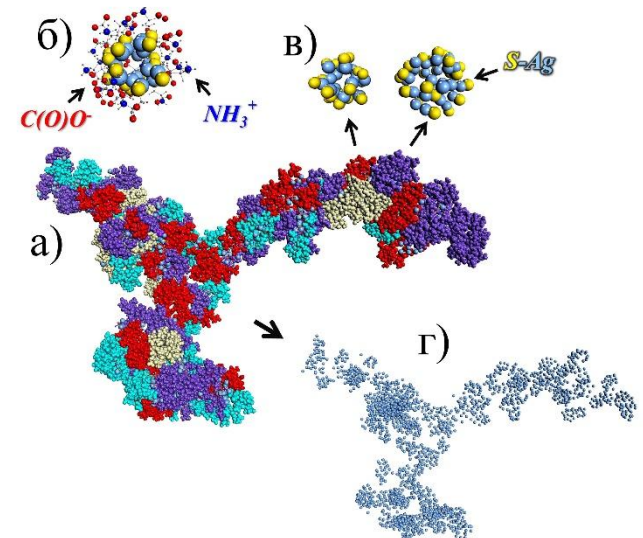
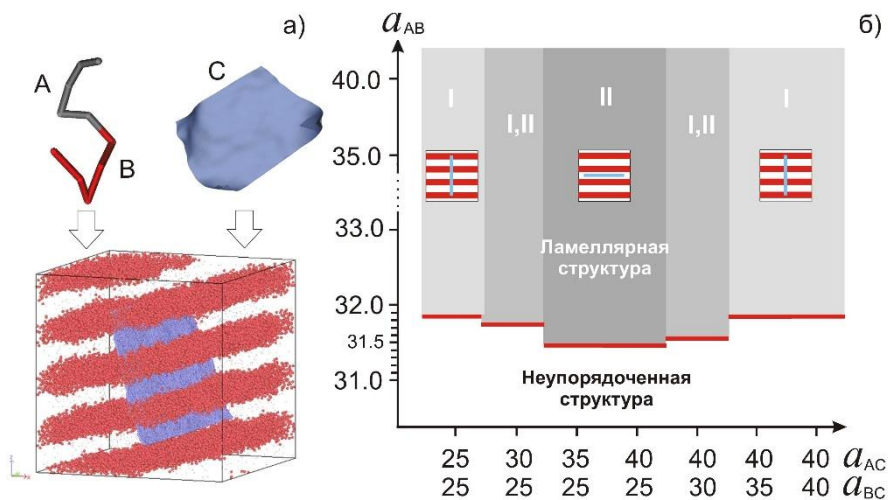
**ЗАДАЧА.** Изучение, характеристика и разработка новых функциональных материалов на основе методологии многомасштабного моделирования.

**СТРАТЕГИЯ.** В данном проекте делается акцент на методы атомистического и мезоскопического компьютерного моделирования такие как атомистическая молекулярная динамика и диссипативная динамика частиц.

**ЦЕЛЬ.** Теоретическая разработка новых подходов к созданию функциональных наноструктурированных материалов.

**ЭФФЕКТ.** Новые функциональные материалы и методы их создания.

**ИСПОЛЬЗОВАНИЕ.** Новые методологии разработки, наноматериалы.



# Моделирование поведения разветвленных молекул, микрогелей и многокомпонентных полимерных систем в растворах и расплавах

Гумеров Р.А.

МГУ имени М.В.Ломоносова

**ЗАДАЧА.** Изучить адсорбционные свойства древовидных молекул, микрогелей на поверхностях, и изменения структурных характеристик пленок блок-сополимеров и нанокомпозитов при набухании.

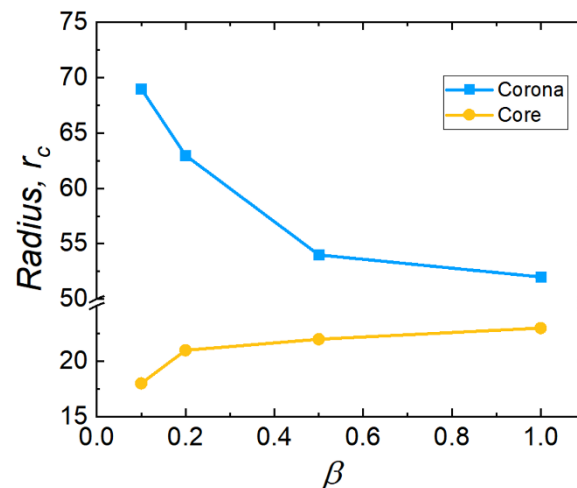
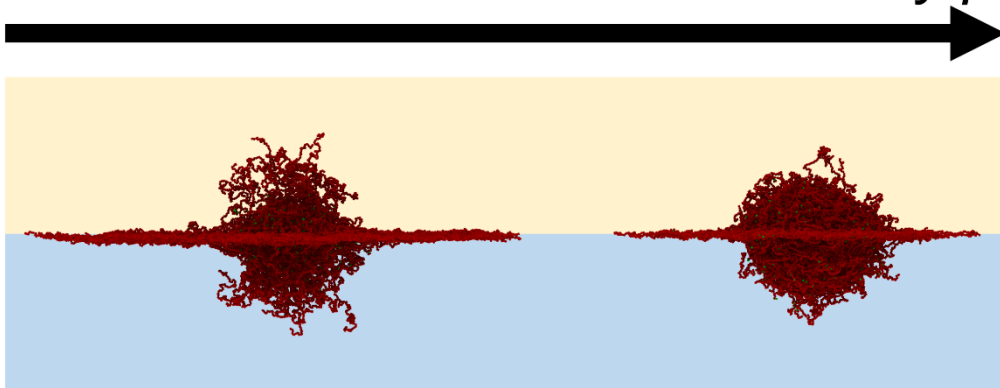
**СТРАТЕГИЯ.** Помещение нескольких молекул в растворитель, наблюдение за их поведением. Поведение моделируется методом диссипативной динамики частиц.

**ЦЕЛЬ.** Добиться стабильных эмульсий и высокоструктурированных пленок. При этом морфология молекул и пленок должна соответствовать определенным ожиданиям.

**ЭФФЕКТ.** Микрогели и древовидные молекулы способны покрывать большую площадь межфазной границы по сравнению с линейными полимерами или ПАВами. Набухшие тонкие пленки блок-сополимеров имеют неоднородное распределение растворителя вдоль доменов.

**ИСПОЛЬЗОВАНИЕ.** Высокие адсорбционные свойства применяются для получения стабильных микро- и наноэмульсий, Тонкие пленки могут быть применены в качестве наноасосов.

Increase of crosslinker reactivity  $\beta$



# Воздействие космической среды на наноструктуры и наноматериалы

Хлебников С.А., Воронина Е.Н., Чирская Н.П.  
НИИЯФ МГУ имени М.В.Ломоносова

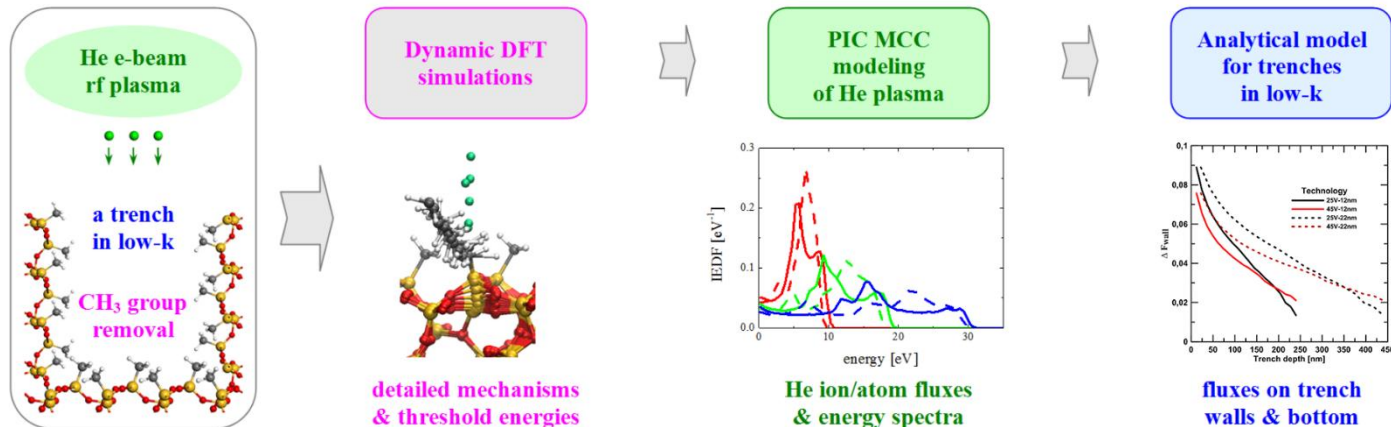
**ЗАДАЧА.** Разработка новых материалов и элементов оборудования космических аппаратов, стойких к воздействию космической среды.

**СТРАТЕГИЯ.** Моделирование изменения свойств наноструктур и наноматериалов под воздействием космической среды методами многомасштабного моделирования.

**ЦЕЛЬ.** Исследование воздействия факторов космического пространства – частиц холодной и горячей магнитосферной плазмы, космической радиации и других – на наноструктуры и наноматериалы на их основе. Использование принципов многомасштабного моделирования к описанию воздействия факторов космического пространства на наноматериалы, выбор набора методов моделирования, подходящих для моделирования воздействия конкретных факторов, а также их сочетаний.

**ЭФФЕКТ.** Прогнозирование поведения наноструктур и наноматериалов в экстремальных условиях космоса. Повышение стойкости материалов космической техники. Выдача рекомендаций по созданию новых материалов, стойких к воздействию космической среды.

**ИСПОЛЬЗОВАНИЕ.** Космическая отрасль, нанотехнологии.





# Рациональное создание ферментов на основе иммуноглобулинов

Головин А.В.

МГУ имени М.В.Ломоносова

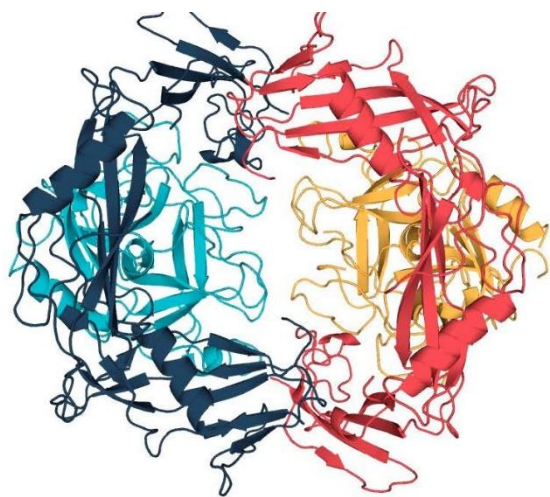
**ЗАДАЧА.** Создание новых ферментов на основе иммуноглобулинов с использованием методов молекулярного моделирования.

**СТРАТЕГИЯ.** Полное исследование пространства мутаций иммуноглобулинов в активном центре и перенос активного из известных ферментов для достижения уровня ферментативной активности сравнимого с природными ферментами.

**ЦЕЛЬ.** Создание вычислительных подходов с использованием методов молекулярной механики, квантовой химии и их гибридов для создания новых ферментов на основе иммуноглобулинов.

**ЭФФЕКТ.** Иммуноглобулины играют важную роль в современной биофармацевтике, придание им каталитических свойств увеличит их эффективность во много раз.

**ИСПОЛЬЗОВАНИЕ.** Медицина, Фармакология, Фундаментальные исследования.



# Новые стратегии многокомпонентного ингибирования митотического веретена для остановки деления опухолевых клеток

Гудимчук Н.Б.<sup>1,2</sup>, Александрова В.В.<sup>1,3</sup>, Ельцов И.А.<sup>1,3</sup>, Лопанская Ю.Н.<sup>1,2</sup>, Федоров В.А.<sup>1,2</sup>, Анисимов М.Н.<sup>1,2</sup>

1) МГУ имени М.В.Ломоносова

2) Центр теоретических проблем физико-химической фармакологии РАН

3) Московский физико-технический институт

**ЗАДАЧА.** Разработка альтернативных противоопухолевых препаратов, которые позволили бы преодолевать устойчивость организма к существующим на данный момент лекарствам, а также их побочные эффекты.

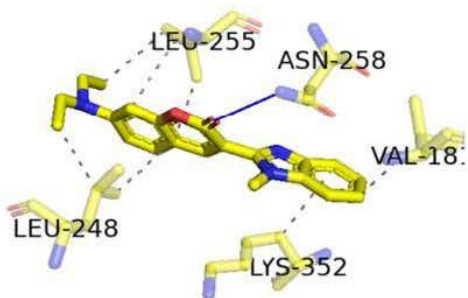
**СТРАТЕГИЯ.** Поиск новых ингибиторов алгоритмом COMPARE по корреляции профилей активности на панели клеток NCI-60.

**ЦЕЛЬ.** Разработка альтернативных противоопухолевых препаратов и создание нового метода работы в данной области.

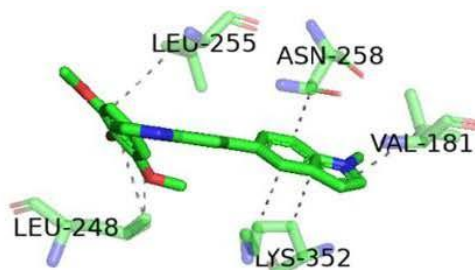
**ЭФФЕКТ.** Научный: создание эффективного метода разработки лекарств внесет вклад в область исследования взаимодействия белков и низкомолекулярных соединений.

**ИСПОЛЬЗОВАНИЕ.** 1) медицина 2) фармакология 3) область науки, занимающаяся взаимодействием белков и низкомолекулярных соединений.

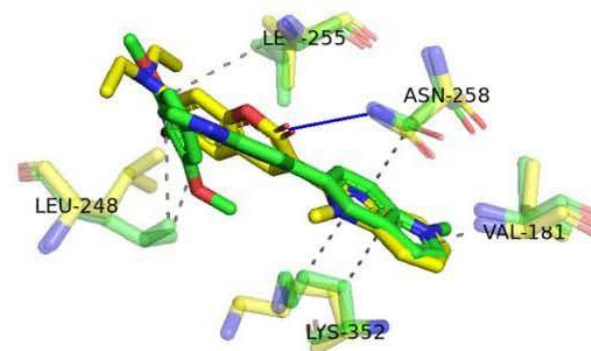
**a)** Coumarin-30



**b)** PDB id: 6LSN



**c)** Overlay



# Моделирование сложных турбулентных течений на неструктурированных сетках

Дубень А.П., Абалакин И.В., Бахвалов П.А., Бобков В.Г.,  
Горобец А.В., Жданова Н.С., Козубская Т.К., Родионов П.В.  
ИПМ им. М.В. Келдыша РАН

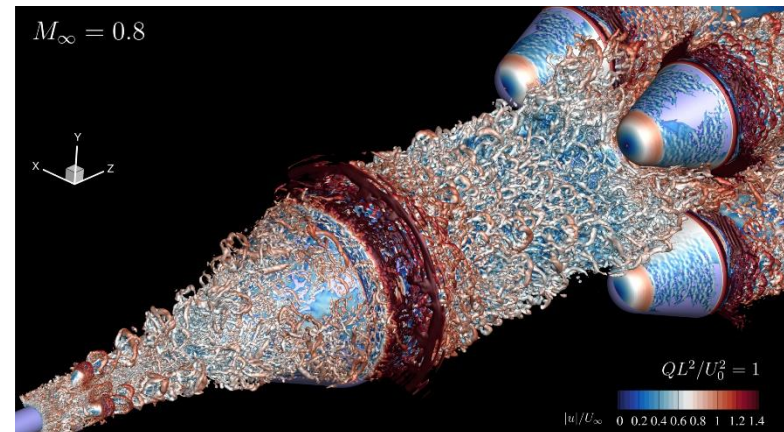
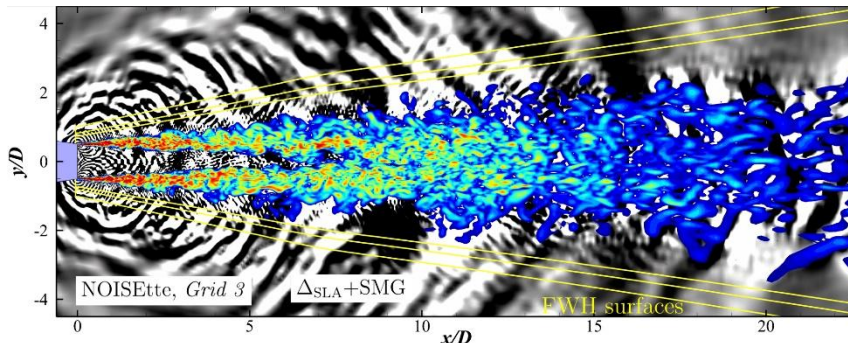
**ЗАДАЧА.** Численное исследование аэродинамических и акустических характеристик плохообтекаемых элементов летательных аппаратов и сложных турбулентных течений.

**СТРАТЕГИЯ.** Проведение численных экспериментов для различных конфигураций, сравнение с экспериментальными данными, выявление оптимальных конфигураций.

**ЦЕЛЬ.** Изучение и улучшение аэродинамических и аэроакустических характеристик элементов летательных аппаратов и сложных турбулентных течений.

**ЭФФЕКТ.** Улучшение аэродинамических свойств летательных аппаратов.

**ИСПОЛЬЗОВАНИЕ.** Авиационная промышленность; вычислительная математика; информатика.



# Теоретическое моделирование электронных и магнитных свойств эндоэдральных фуллеренов и их производных

Иоффе И.Н., Луконина Н.С., Мазалева О.Н., Сударькова С.М., Пыхова А.Д., Хиневич В.Э.  
МГУ имени М.В.Ломоносова

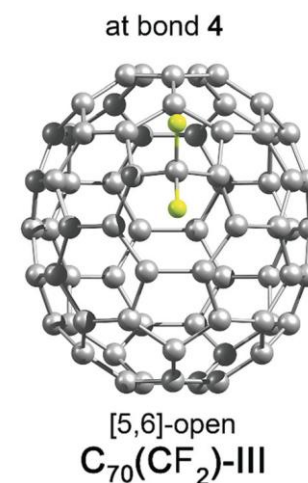
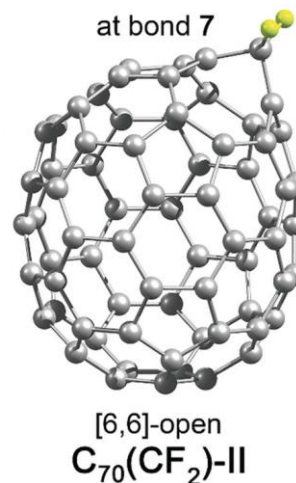
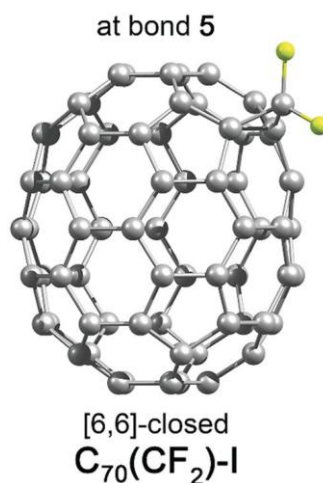
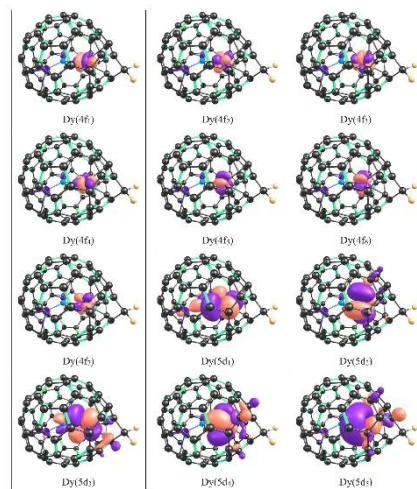
**ЗАДАЧА.** Данный проект посвящен поиску перспективных молекулярных магнетиков на основе эндоэдральных металлофуллеренов и отработке теоретических подходов к их описанию.

**СТРАТЕГИЯ.** Изучение основных состояний - динамики эндоэдральных атомов внутри углеродного каркаса и ее взаимосвязи с мотивами внешнесферной функционализации.

**ЦЕЛЬ.** Выяснение особенностей динамики эндоэдральных кластеров внутри различных углеродных каркасов и связи ее с региохимическими особенностями внешнесферной функционализации.

**ЭФФЕКТ.** Проект позволит лучше понять фундаментальные аспекты электронных и магнитных свойств эндоэдральных фуллеренов, а также предложить такие эндоэдральные соединения, которые могут найти применение в новых перспективных материалах для спинтроники и хранения информации.

**ИСПОЛЬЗОВАНИЕ.** Фундаментальное и прикладное материаловедение.





# Молекулярное моделирование биохимических и биофизических процессов в холинэстеразах и разработка лекарственных препаратов, взаимодействующих с ними

Лущеккина С.В., Новичкова Д.А., Капуста Д.П., Коц Е.Д., Бойко К.М.

Институт биохимической физики им. Н.М. Эмануэля РАН

МГУ имени М.В.Ломоносова

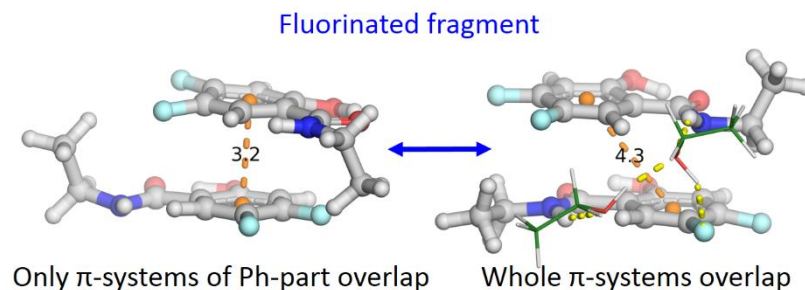
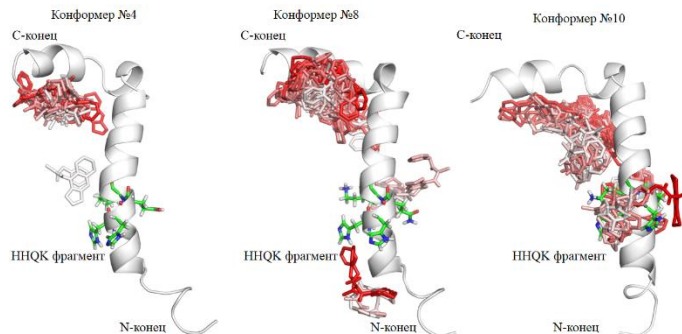
**ЗАДАЧА.** Холинэстеразы играют ключевую роль в функционировании человеческого организма. Исследование их биохимических и биофизических свойств необходимо для более глубокого понимания физиологии человека. Холинэстеразы вовлечены в развитие различных серьезных заболеваний (болезнь Альцгеймера, миастения), и являются мишенями для лекарственной терапии этих заболеваний. Также они являются мишенями отравляющих веществ, и исследование механизмов их функционирования необходимо для разработки защитных средств.

**СТРАТЕГИЯ.** Молекулярное моделирование динамики, конформационных переходов в белковых молекулах ферментов, исследование энергетических профилей реакций ферментов с ингибиторами и субстратами с использованием методов молекулярного докинга, молекулярной динамики, квантовой механики, комбинированного метода квантовой механики и молекулярной механики, комбинированного метода квантовой механики и молекулярной динамики.

**ЦЕЛЬ.** Получение информации о процессах, протекающих в биологических системах и об их взаимодействии с различными химическими соединениями -- отравляющими веществами и лекарственными препаратами. Разработка новых лекарственных препаратов и защитных средств.

**ЭФФЕКТ.** Расширение фундаментальной базы биохимических и биомедицинских исследований. Разработка и внедрение новых лекарственных препаратов.

**ИСПОЛЬЗОВАНИЕ.** Фармакология, медицина. Препараты для терапии болезни Альцгеймера, миастении, защитных средств для авиации (пассажиры, команда, наземный персонал) от аэротоксического синдрома.



# Структурные и динамические свойства полимерных нанокompозитных материалов

Люлин С.В., Гуртовенко А.А., Ларин С.В., Фалькович С.Г.,  
Назарычев В.М., Волгин И.В., Глова А.Д.

Институт высокомолекулярных соединений РАН

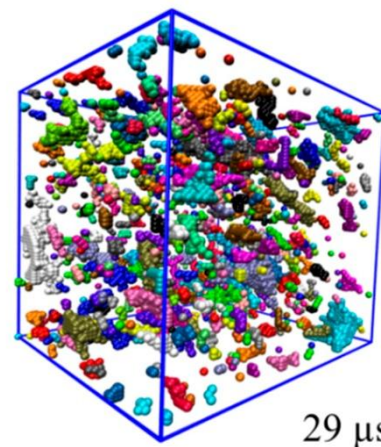
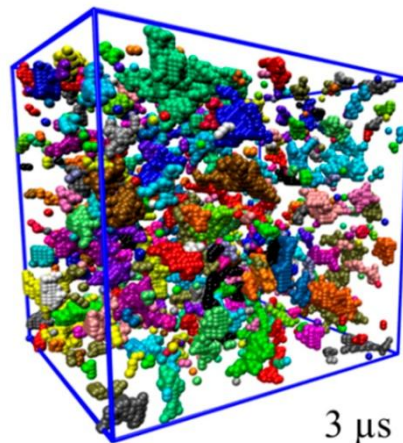
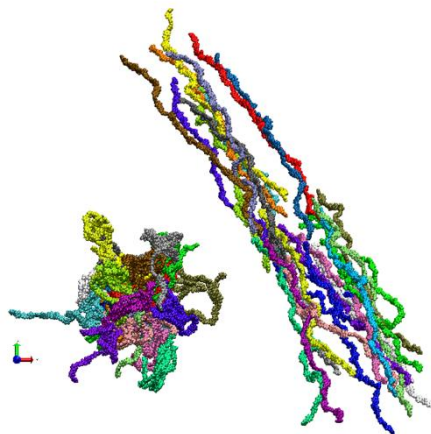
**ЗАДАЧА.** Изучение и прогнозирование структурных и динамических свойств полимерных композиционных материалов с наноразмерным наполнителем.

**СТРАТЕГИЯ.** Разработка моделей и компьютерное моделирование композиционных материалов с использованием методов квантовой химии и методов молекулярной динамики.

**ЦЕЛЬ.** Разработать модель для описания композиционных материалов, позволяющую с высокой степенью достоверности определять структурные и динамические свойства полимерных композиционных материалов в процессе компьютерного моделирования. Изучить влияние на свойства материалов структуры полимера, образующего матрицу, а также типа, формы и размеров наполнителя.

**ЭФФЕКТ.** Метод расчета и прогнозирования свойств полимерных композиционных материалов.

**ИСПОЛЬЗОВАНИЕ.** Разработка и создание новых полимерных композиционных материалов с наноразмерным наполнителем.



# Фундаментальные исследования процессов горения и детонации применительно к развитию основ энерготехнологий

Левин В.А., Мануйлович И.С., Журавская Т.А., Сутырин О.Г.  
МГУ имени М.В.Ломоносова

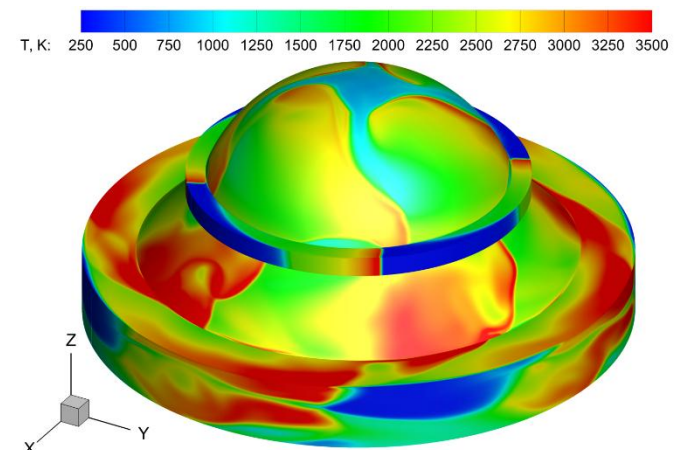
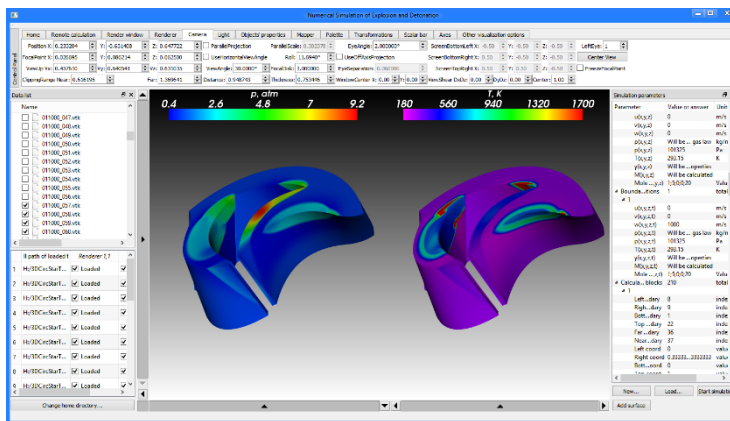
**ЗАДАЧА.** Изучение динамики и структуры волн детонации в неоднородных смесях в каналах сложной геометрии и камерах сгорания на основе разработанных математических моделей и оригинального программного комплекса.

**СТРАТЕГИЯ.** Быстрые параллельные расчеты течений реагирующих многокомпонентных газовых смесей в областях сложной геометрии.

**ЦЕЛЬ.** Создание оригинального программного комплекса, предназначенного для моделирования 1D-, 2D- и 3D-течений.

**ЭФФЕКТ.** Быстрое и простое моделирование сложных 1D-, 2D- и 3D-течений многокомпонентных реагирующих газовых смесей с горением и детонацией.

**ИСПОЛЬЗОВАНИЕ.** Разработка новых типов двигателей, решение проблем, связанных с взрывобезопасностью на различных объектах.





# Суперкомпьютерное моделирование биомолекулярных систем на основе квантово-химических и молекулярно-динамических методов: от ферментативного катализа до оптогенетики

Немухин А.В.

МГУ имени М.В.Ломоносова

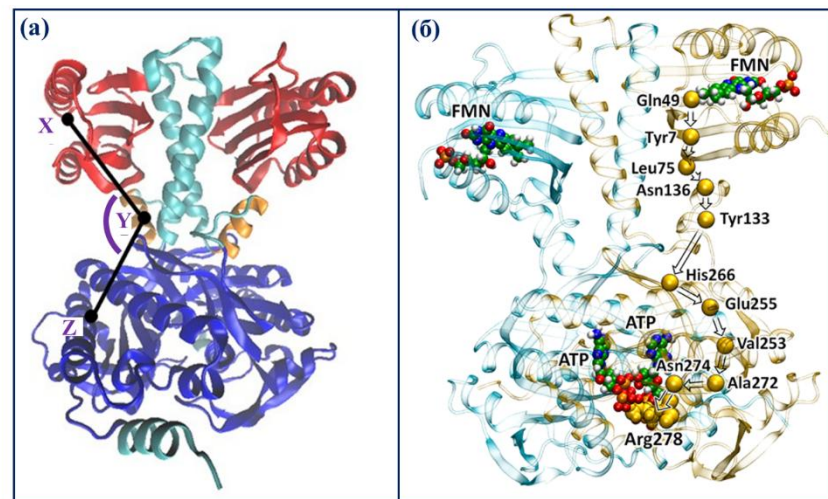
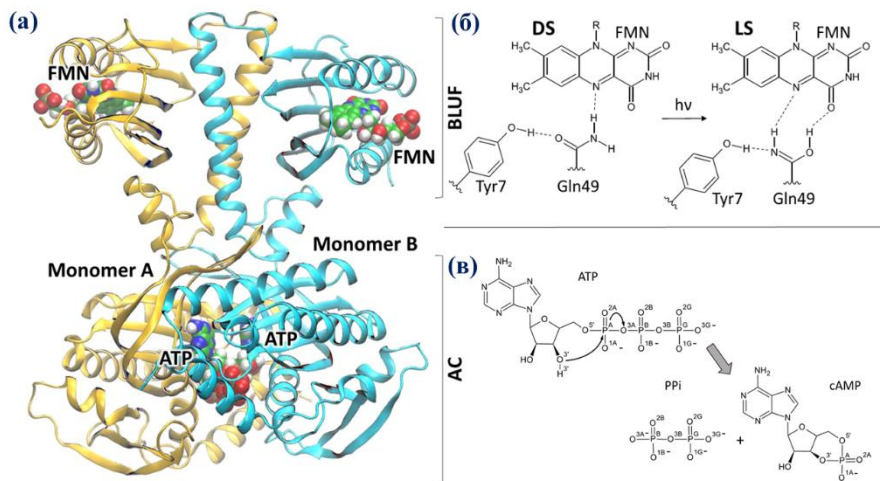
**ЗАДАЧА.** Развитие и применение методов моделирования строения и свойств биомолекулярных систем.

**СТРАТЕГИЯ.** Практическая реализация комбинированного метода квантовой механики/молекулярной механики (КМ/ММ) позволяет выйти на качественно новый уровень в компьютерном моделировании свойств биомолекулярных систем.

**ЦЕЛЬ.** Решение конкретных задач моделирования биомолекул с использованием методов квантовой химии, КМ/ММ и МД, ориентированных на суперкомпьютерные расчеты.

**ЭФФЕКТ.** Будут исследованы молекулярные механизмы действия биомолекулярных систем, необходимые для разработки способов управления процессами в этих системах.

**ИСПОЛЬЗОВАНИЕ.** Биомедицина, биотехнология.



# Постгеномные исследования

Кутов Д.К., Ильин И.С., Тащилова А.С., Сулимов В.Б., Сулимов А.В.  
МГУ имени М.В.Ломоносова

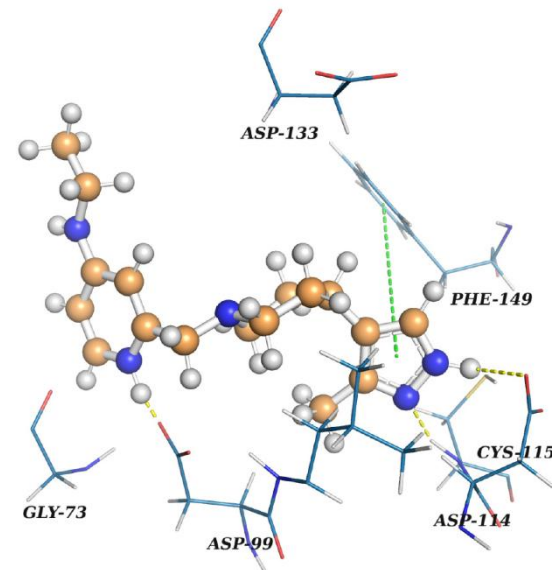
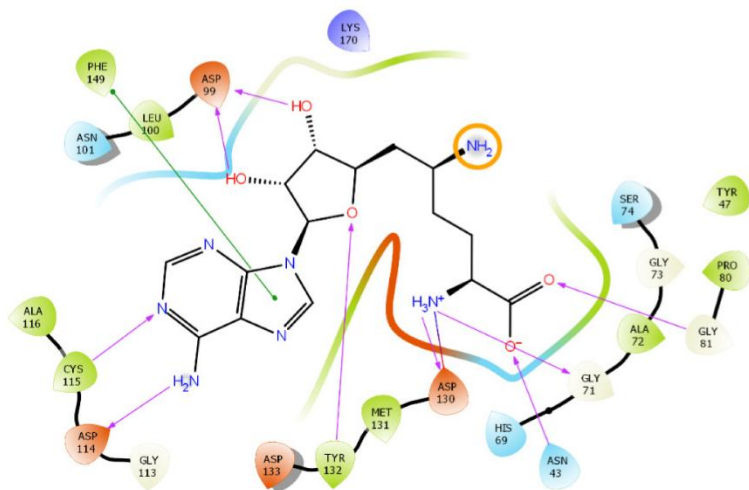
**ЗАДАЧА.** Разработка новых ингибиторов урокиназы, методами молекулярного моделирования.

**СТРАТЕГИЯ.** Виртуальный скрининг баз данных лигандов методами докинга, квантовой химии и молекулярной динамики для оценки энергий взаимодействия белок-лиганд.

**ЦЕЛЬ.** Создание методики, основанной на молекулярном моделировании с высокой точностью расчетов, позволяющей эффективно предсказывать ингибиторы белков, в частности ингибиторы урокиназы.

**ЭФФЕКТ.** Эффективная, быстрая и недорогая компьютерная методика разработки новых лекарств, а также новые ингибиторы урокиназы.

**ИСПОЛЬЗОВАНИЕ.** Разработанные методы позволят предсказывать новые ингибиторы для заданных белков-мишеней, в частности, для урокиназы, а новые ингибиторы урокиназы могут стать основой для новых лекарств.



# Моделирование динамики хроматина

Армеев Г.А., Косарим Н.А., Горковец Т.К., Новоселецкий В.Н.,  
Поспелова Ю., Андреева Е., Князева А., Шайтан А.К.  
МГУ имени М.В.Ломоносова

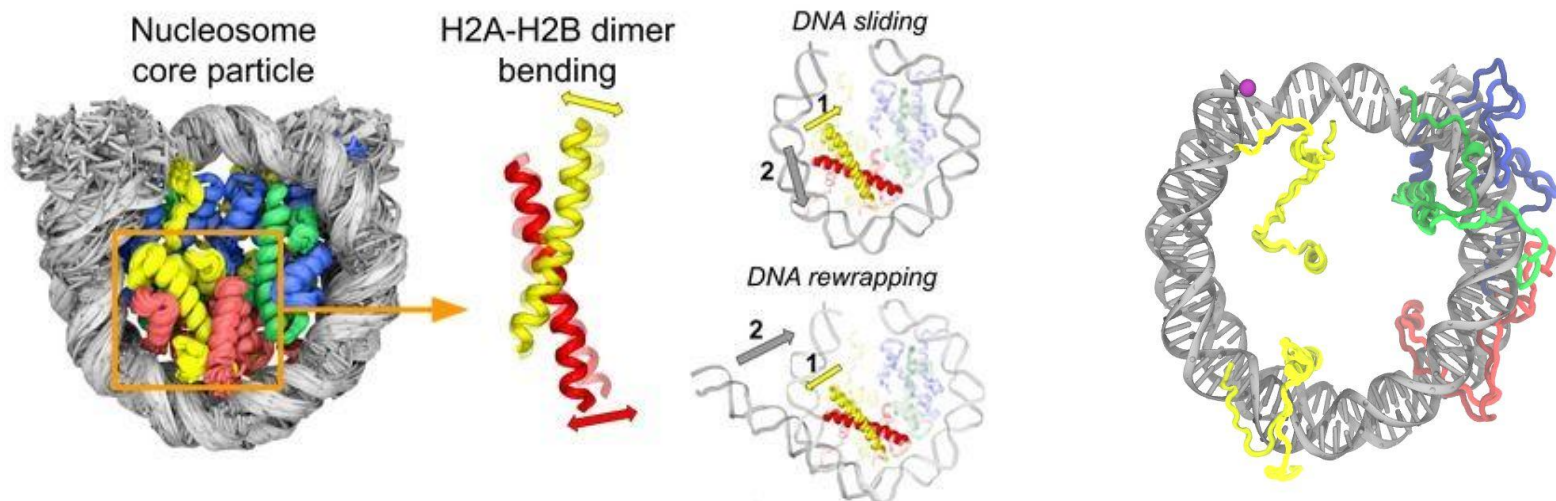
**ЗАДАЧА.** Понимание организации и динамики белок-белковых и ДНК-белковых взаимодействий в ядре эукариотической клетки.

**СТРАТЕГИЯ.** Молекулярное моделирование (молекулярная динамика, белок-белковый докинг, решение уравнение Пуассона-Больцмана) взаимодействий белков и ДНК в хроматине.

**ЦЕЛЬ.** Осуществить комплексное рассмотрение основных взаимодействий белков и ДНК на уровне элементарных частиц хроматина - нуклеосом. Выявить влияние различных взаимодействий и модификаций на стабильность и конформацию нуклеосом, и, как следствие, на регуляцию функционирования живых организмов.

**ЭФФЕКТ.** Выявление новых целей для разработки лекарств.

**ИСПОЛЬЗОВАНИЕ.** Медицина, биология, биотехнология, эпигенетика.



# Молекулярное моделирование биокаталитических процессов

Балдин С.М., Копылов К.Е., Дробот В.В., Бочкова А.А.,  
Кирилин Е.М., Подшивалов Д.Д., Швядас В.К.  
МГУ имени М.В.Ломоносова

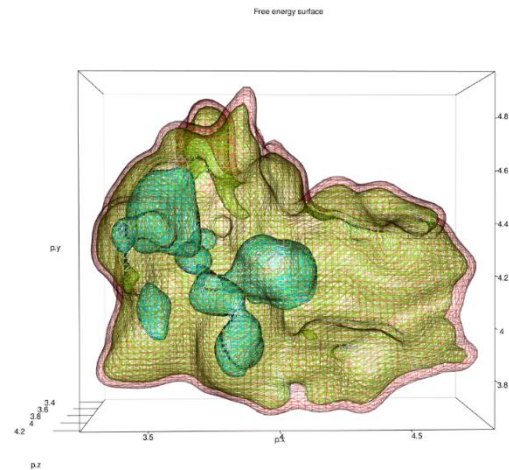
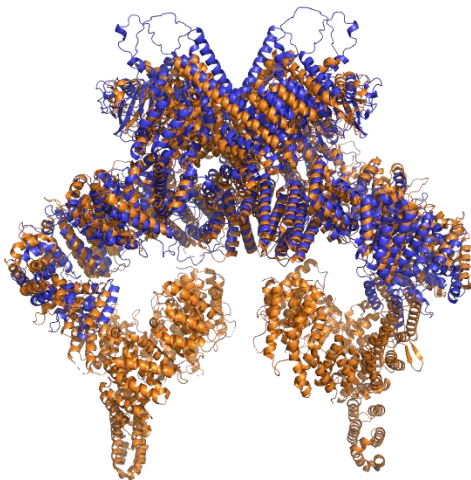
**ЗАДАЧА.** Получение детальной информации о химическом механизме работы ферментов.

**СТРАТЕГИЯ.** Использование оригинальной комплексной методологии молекулярного моделирования с применением молекулярного докинга, классической молекулярной динамики, метадинамики и гибридных квантово-механических / молекулярно-механических подходов (QM / MM).

**ЦЕЛЬ.** Изучение механизмов действия ферментов, возможностей модификации их физико-химических свойств и ингибирования.

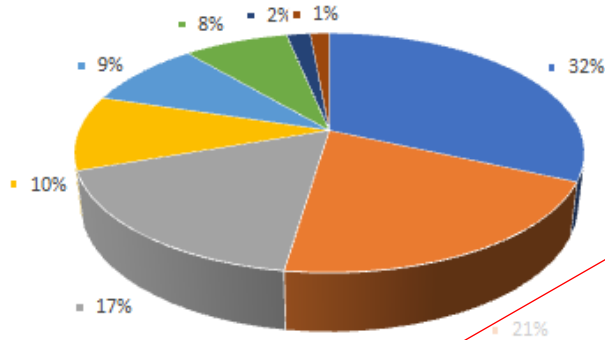
**ЭФФЕКТ.** Улучшение понимания химического механизма функционирования ферментов, динамики каталитического процесса в активных центрах и её связи с функцией.

**ИСПОЛЬЗОВАНИЕ.** Биохимия, биоинженерия, энзимология, физическая химия.

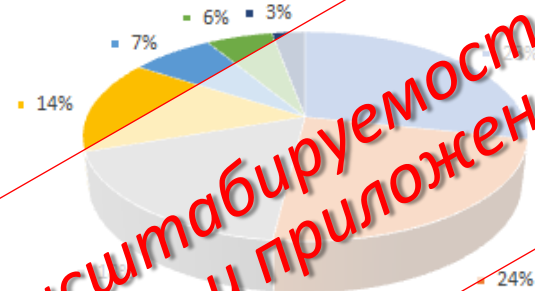


# Суперкомпьютерный комплекс Московского университета (краткая статистика за 2022 год)

Распределение проектов по приоритетам Стратегии НТП



- Индустрия наносистем
- Науки о жизни
- Информационно-телекоммуникационные системы
- Транспортные и космические системы
- Энергоэффективность, энергосбережение, ядерная энергетика
- Рациональное природопользование, экология
- Безопасность и противодействие терроризму
- Перспективные виды вооружения, космос и специальная техника



Распределение проектов по областям науки

- Химия
- Физика и астрономия
- Математика, информатика, механика
- Биология и медицинская наука
- Информационные технологии и вычислительные системы
- Науки о земле
- Фундаментальные основы инженерных наук

**Степень параллельности и масштабируемость –  
ключевые свойства алгоритмов и приложений**

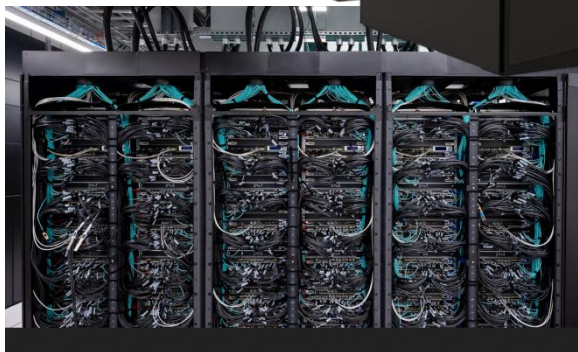
Пользователи – 575  
Проекты – 200  
Гранты РНФ, РФФИ – более 80  
Проекты Минобрнауки – более 20  
Публикации – 240  
Диссертации – 16

Подразделения МГУ – 72%  
Внешние организации – 28%

Проектов РНФ-ОИ-2022 – 8 (максимум  
среди объектов научной  
инфраструктуры РНФ)

# Суперкомпьютер Frontier, США

(#1 Top500 в 2022 г.)



9 408 вычислительных узлов,  
в каждом узле:  
1 x CPUs (AMD "Trento", 64 ядра, 2GHz)  
4 x GPU (AMD Radeon MI250X)  
8 730 112 ядер

Производительность:  
Пик (теория): **1.68 Eflop/s**  
Тест Linpack: **1.1 Eflop/s** (65%)

Оперативная память = 9.2 PBytes  
HDD = 716 PB (+37 PB на узлах)

21 MW (всего 29 MW) – 52.2 Gflops/Watt  
(1 стойка – 62.68 Gflops/Watt)

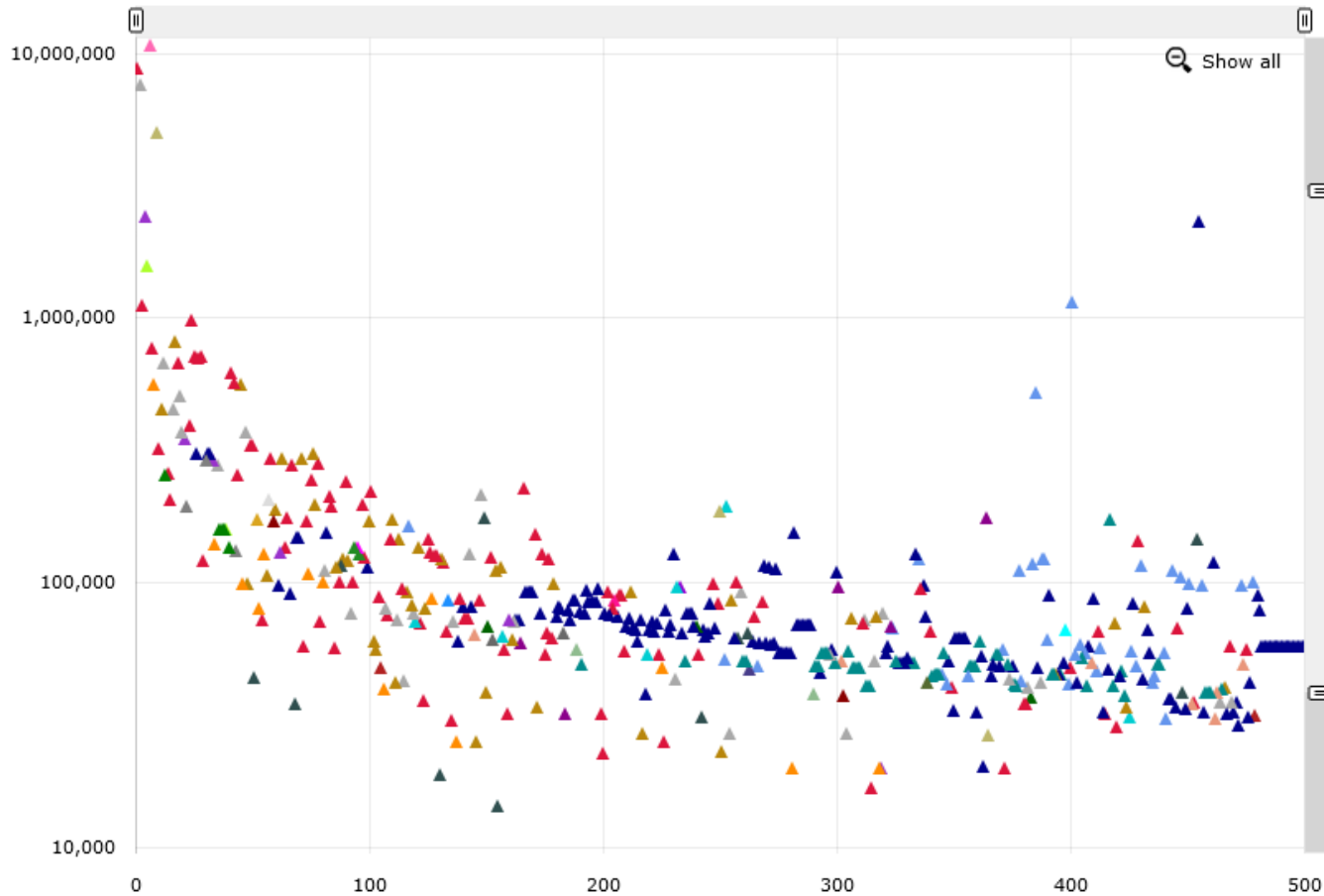
#1 – Top500  
#1 – Green500  
#1 – HPL-AI

# Годы, флопсы и степень параллелизма

(когда и как был достигнут очередной 'X'flops)

$10^6$	Mflops	1964 г.	CDC 6600	10 MHz	1 CPUs
$10^9$	Gflops	1985 г.	Cray 2	125 MHz	8 CPUs
$10^{12}$	Tflops	1997 г.	ASCI Red	200 MHz	9152 CPUs
$10^{15}$	Pflops	2008 г.	Roadrunner	3,2 GHz	122400 Cores
$10^{18}$	Eflops	2022 г.	Frontier	2,0 GHz	8730112 Cores

# Число ядер в системах списка Top500 (<http://top500.org>, июнь, 2022 г.)



Позиция системы в списке Top500



# Доступный параллелизм: NVIDIA GPU (Tesla V100, Volta)



Параллельный мир вокруг нас...

- 5120 ядер,
- 7.5 Тфлопс на двойной точности,
- 15 Тфлопс на одинарной точности,
- 120 Тфлопс (tensor flops, FP16)

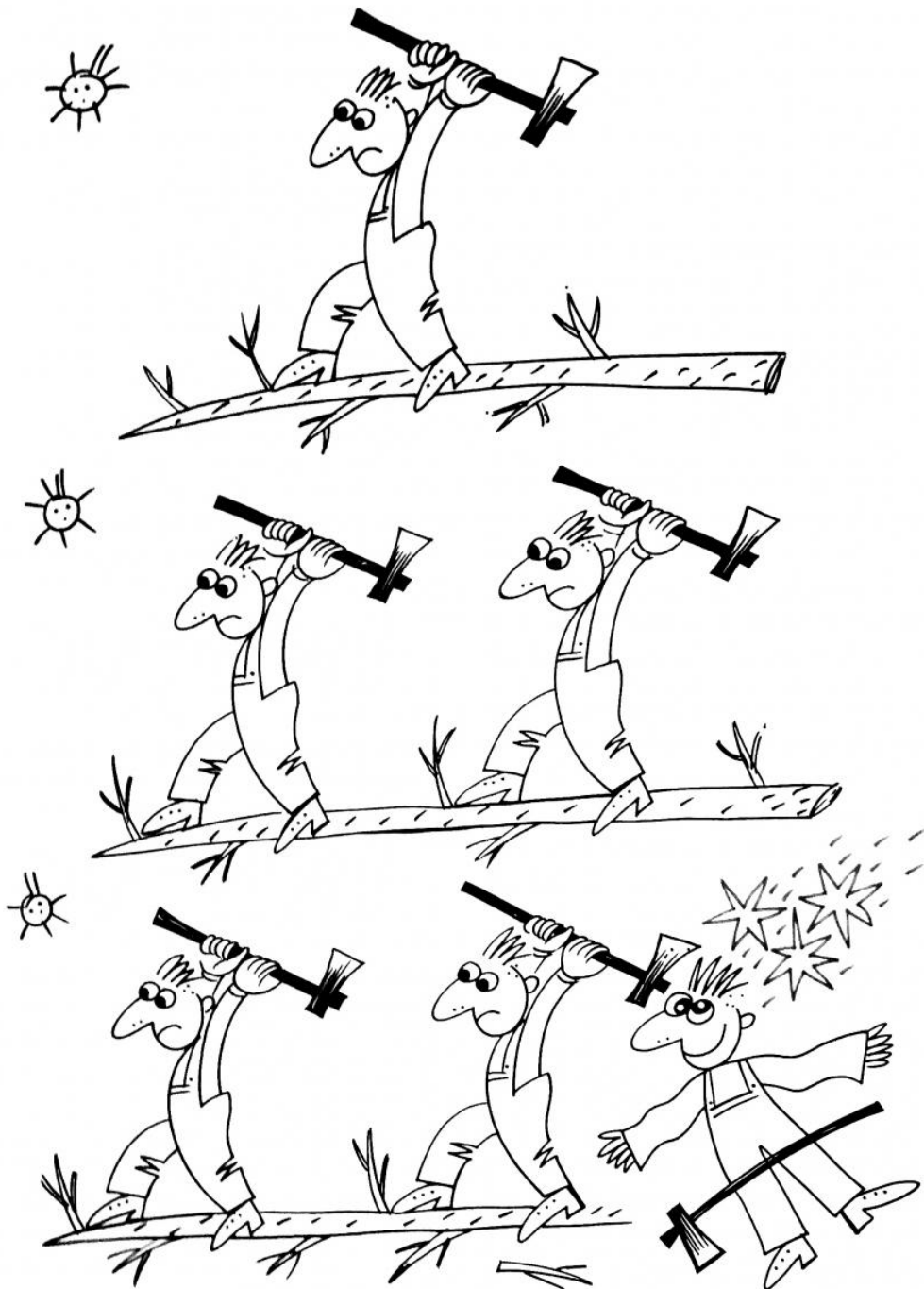


На данном изображении показана фотография пункта оплаты трассы М11 (фото с иллюминатора самолета). Пункт оплаты в данном случае является своеобразным bottleneck-ом на трассе, так как автомобили должны существенно снизить скорость, чтобы произвести оплату. Чтобы не создавать пробки, пункты оплаты “распараллелены”. За счет этого средняя скорость прохождения данного участка остается высокой. Можно сказать, что каждый из процессов движения (один туда, другой обратно) разбивается внутри себя на большее количество нитей (потокков движения) перед узким местом, а после его прохождения возвращается в более выгодное с точки зрения ресурсов меньшее количество нитей, т.к. на дальнейшем участке дороги большое распараллеливание не принесет выигрыша в производительности из-за движения на пределе ограничения по скорости.



На данной фотографии мы можем увидеть, что несколько сурикатов смотрят в одну сторону, а несколько в другую. Тем самым они разделяют между собой контроль над территорией, поделив ее на несколько подконтрольных частей. Если бы за наблюдением отвечал один сурикат, то выполнение контроля за территорией подразумевало бы постоянную смену позиции (что напоминает смену контекстов). Такое действие затрачивает много энергетических ресурсов. Тем временем если параллельное наблюдение осуществляют несколько сурикатов, которые не требуют между собой постоянной передачи информации (а только сигнал об опасности), качество

работы увеличивается. Таким образом сурикаты реализуют параллельную обработку данных (об их окружении), разделяя между собой всю полученную информацию о местности.



Попытки повышения качества параллельных вычислений по одному из показателей (ускорению или эффективности) может привести к ухудшению ситуации по другому показателю, ибо показатели качества параллельных вычислений являются противоречивыми. При увеличении числа процессоров эффективность распараллеливания, как правило, уменьшается из-за роста накладных расходов. На картине это иллюстрируется в шуточной форме.



Дедлок.



Иллюстрация ситуации дедлока в классическом произведении.



Данная картинка иллюстрирует ситуацию дедлока при объяснении ситуации дедлока.



Параллелизм присутствует во многих сферах нашей жизни, однако не везде он уместен. При вождении, например, параллелизм даже опасен.



При работе множества сливов вода сбрасывается вниз быстрее и эффективнее, что демонстрирует многопоточный (также и в прямом смысле этого слова) параллелизм.





Многопоточный чайник. Иногда приходится менять архитектуру программы для увеличения степени параллелизма (с обычным чайником такая параллельная работа невозможна).



В командной гребле очень важна синхронная работа. В идеальном исполнении командная байдарка обгонит одиночную байдарку (как обычно параллельные программы обгоняют линейные), но если гребцы будут мешать друг другу, цепляться веслами (лезть в чужую область памяти, дедлочить), то командная байдарка не сдвинется с места, может даже перевернуться, когда как одиночная уйдет спокойно дальше.



Изображения символизируют классификацию Флинна по наличию параллелизма в потоках команд и данных. Музыкальный инструмент символизирует данные, а игра на нем - инструкцию. (SISD - исполнитель, играющий определенную мелодию на своем музыкальном инструменте. MISD - на одной и той же гитаре два исполнителя играют разные ноты и разные мелодии. SIMD - оркестр исполнителей, играют одно и то же, но на разных гитарах. MIMD - несколько исполнителей играют каждый свою отличающуюся партию на своем инструменте).

Хм, студентов  
слишком много,  
надо что-то  
придумать!  
А что, если...



**Зачет в форме  
беседы с  
преподавателем**

# ЗАЧЕТ



**Зачет в форме  
электронного  
тестирования**

...масштаби-  
ровать систему  
зачета вширь.



Или  
даже  
вглубь!



Масштабируемся вглубь и вширь  
на примере формы зачета.  
Систему зачета, при которой один  
преподаватель лично беседует с  
каждым студентом,  
«масштабируема вширь», так как  
можно увеличить количество  
преподавателей, принимающих  
зачет в форме личной беседы (то  
есть увеличить  
производительность системы  
(быстрее принять зачет) за счет  
добавления дополнительных  
программных или аппаратных  
средств), и «масштабируема  
вглубь», так как можно провести  
зачет в форме электронного  
тестирования (то есть заменить  
форму принятия зачета  
(платформу) на новую,  
обладающую большей  
производительностью).

Параллельный мир вокруг нас...

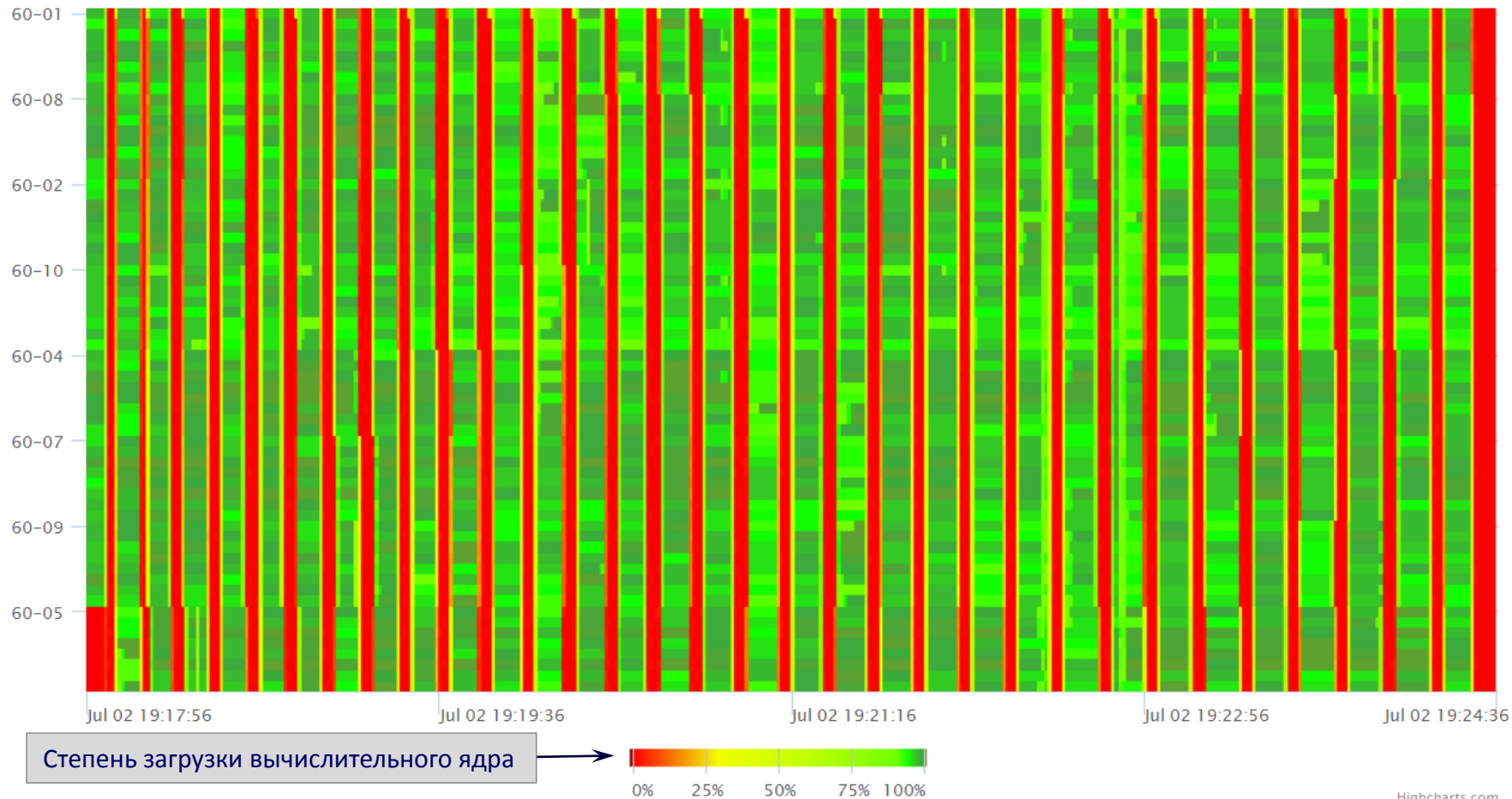
Параллельный компьютерный мир вокруг нас...

Не простой параллельный компьютерный мир вокруг нас...

```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#define size 5, MPI_COMM_WORLD);
#define REP 5, MPI_COMM_WORLD);
#define LOOP 1024
#define ANY_SOURCE 5, MPI_COMM_WORLD);
#define ANY_SIZE 10000
#define ANY_SOURCE 5, MPI_COMM_WORLD);
int main (int argc, char ** av)
{
    int n;
    #pragma omp parallel private (n)
    {
        #pragma omp for
        for (n=1; n<=size; n++)
        {
            MPI_Send(&rbuf1, 1, MPI_FLOAT, MPI_ANY_SOURCE, 5, MPI_COMM_WORLD);
            MPI_Send(&rbuf2, 1, MPI_FLOAT, MPI_ANY_SOURCE, 5, MPI_COMM_WORLD);
            MPI_Recv(&rbuf1, 1, MPI_FLOAT, MPI_ANY_SOURCE, 5, MPI_COMM_WORLD);
            MPI_Recv(&rbuf2, 1, MPI_FLOAT, MPI_ANY_SOURCE, 5, MPI_COMM_WORLD);
        }
    }
    printf("Number of threads = %d\n", omp_get_num_threads());
    return 0;
}
```

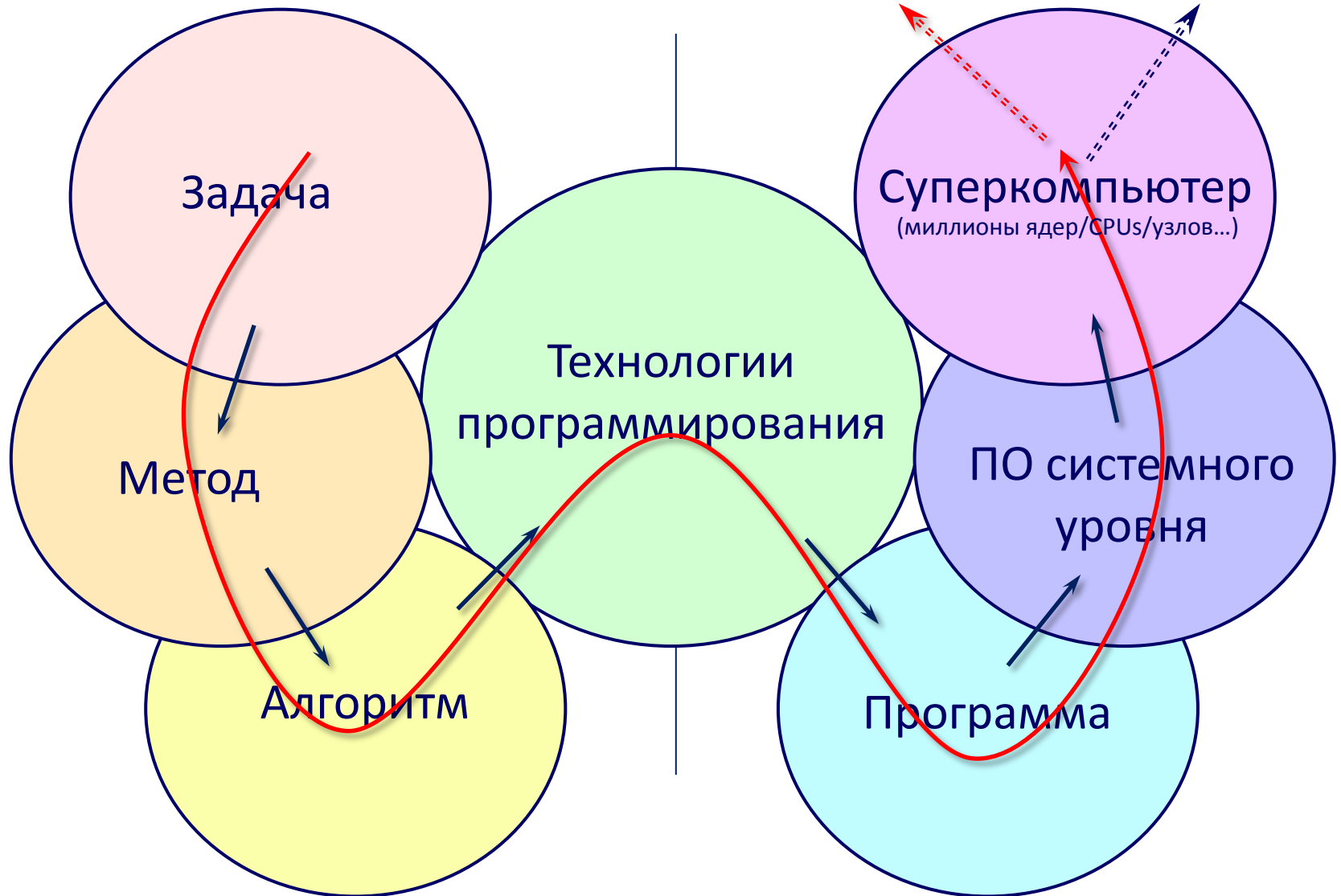
Ожидаемый эффект от одновременной (параллельной) работы:  
один сделает быстро, двое - быстрее, трое – еще быстрее...

# Степень параллельности и масштабируемость – ключевые свойства алгоритмов и приложений



# Решение задач и вычислительные системы

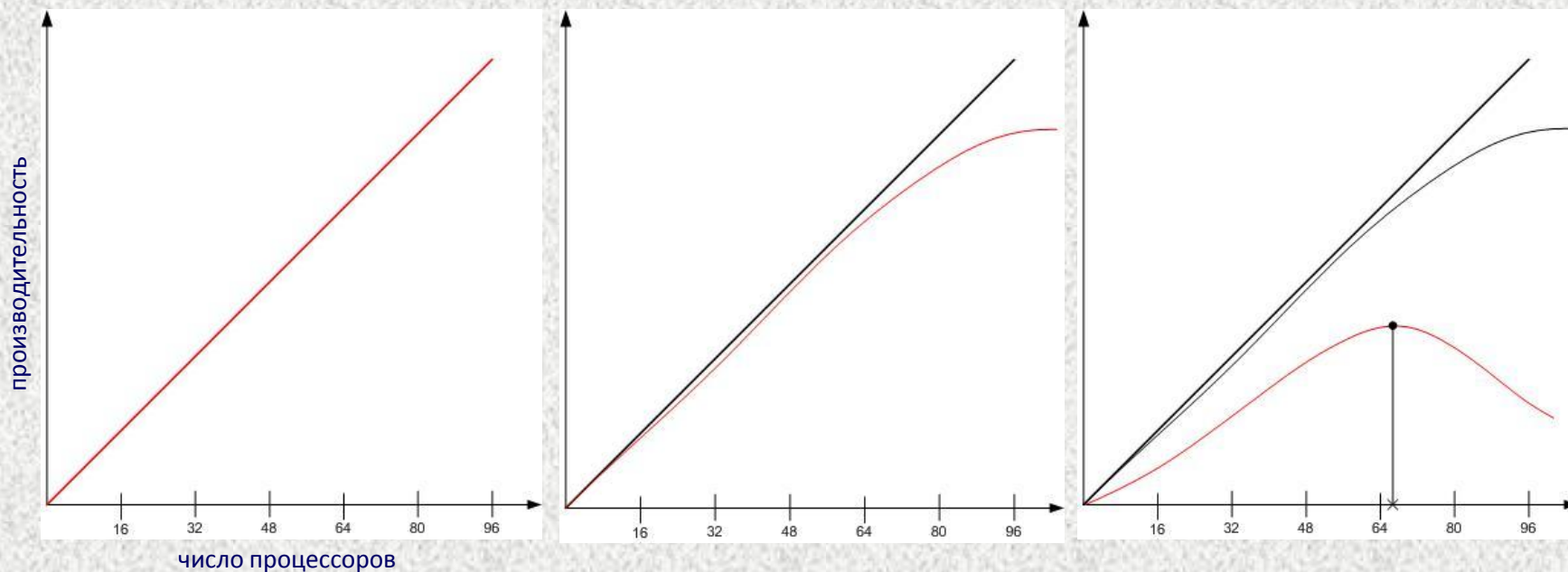
(Реальность) Реальная производительность  $\xleftrightarrow{\text{Огромный разрыв!}}$  (Ожидания) Пиковая производительность



Алгоритмическая сторона

Компьютерная сторона

# 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 центр сегодня

---

- 
- A photograph of a supercomputer center. The image shows a long, curved aisle between rows of dark server racks. The racks are densely packed with components, and numerous cables are visible running along the top of the racks. The floor is light-colored, and the overall environment is clean and industrial. A semi-transparent text box is overlaid on the center of the image, containing a bulleted list of characteristics of supercomputers.
- Огромный потенциал,
  - Высокая технологическая сложность,
  - Сложное сопровождение,
  - Высокая стоимость,
  - Большое разнообразие архитектур,
  - Сложность современных архитектур.

# Суперкомпьютерный центр сегодня

## (Суперкомпьютер Ломоносов)

Пиковая производительность	1.7 Pflops	
Производительность на Linpack	901.9 Tflops	
Эффективность	53 %	
Вычислительные узлы (Intel, x86)	5 104	
Вычислительные узлы (GPU)	1 065	
Процессоров Intel Xeon (X5570/X5670)	12 346	
Процессоров GPU (NVIDIA X2070)	2 130	
Ядер x86, всего	52 168	
Ядер GPU, всего	954 240	
Оперативная память	92 TBytes	
Коммуникационная сеть	QDR 4x Infiniband / 10 GbE	
Система хранения данных	1.75 Pbytes, Lustre, NFS, ...	
Операционная система	Clustrx T-Platforms Edition	
Площадь (вычислитель)	252 м <sup>2</sup>	
Энергопотребление	2.7 MW	

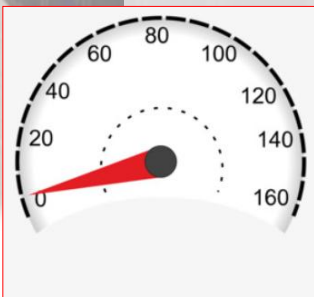
Компоненты  
стека ПО

Сложная  
иерархия памяти

Пользователи

Проекты

...



← скорость будет такой будет, если оставить суперкомпьютер без должного управления...

# Обеспечение качества работы СКЦ

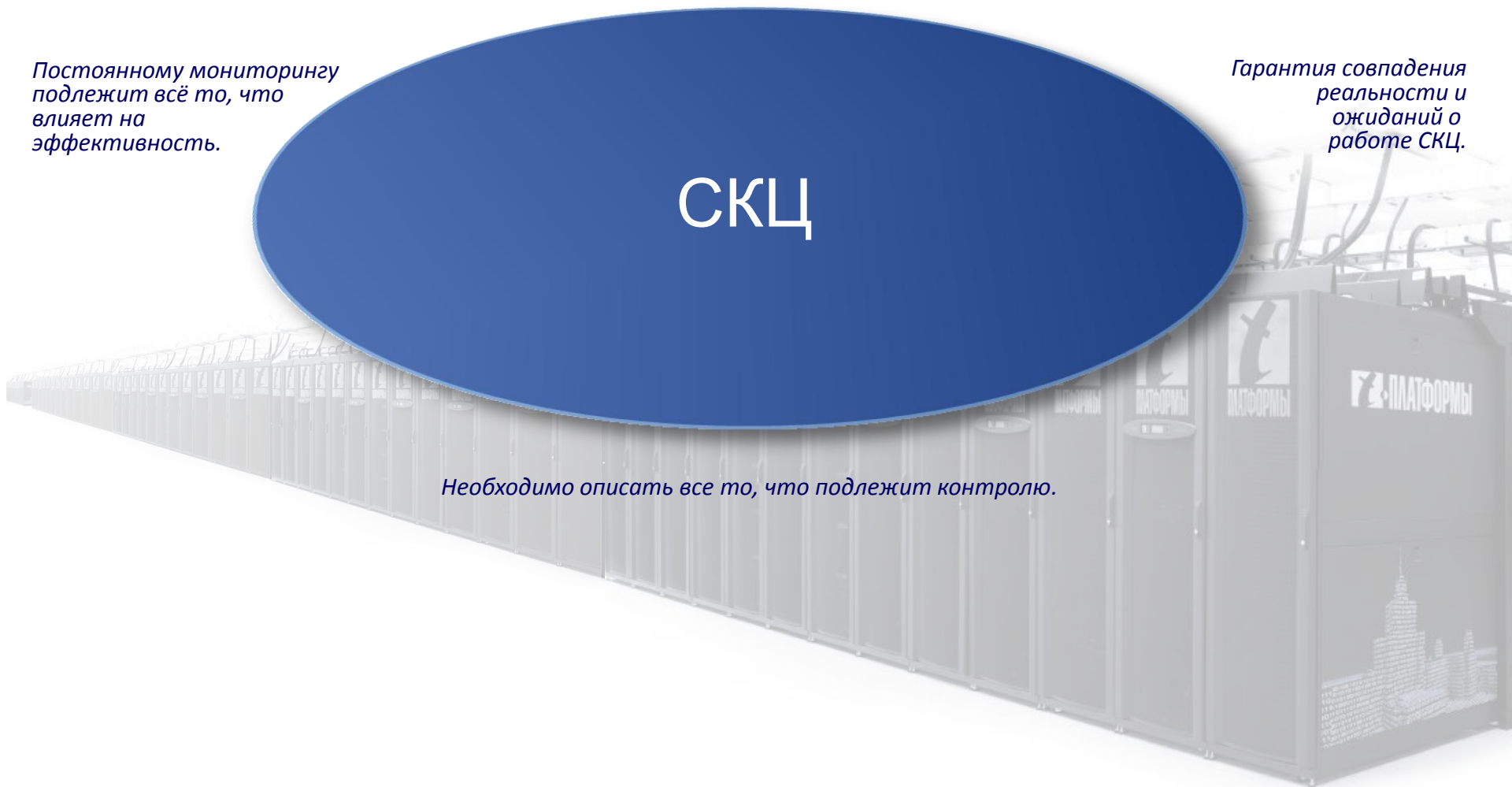
---

*Постоянному мониторингу  
подлежит всё то, что  
влияет на  
эффективность.*

СКЦ

*Гарантия совпадения  
реальности и  
ожиданий о  
работе СКЦ.*

*Необходимо описать все то, что подлежит контролю.*



# Обеспечение качества работы СКЦ

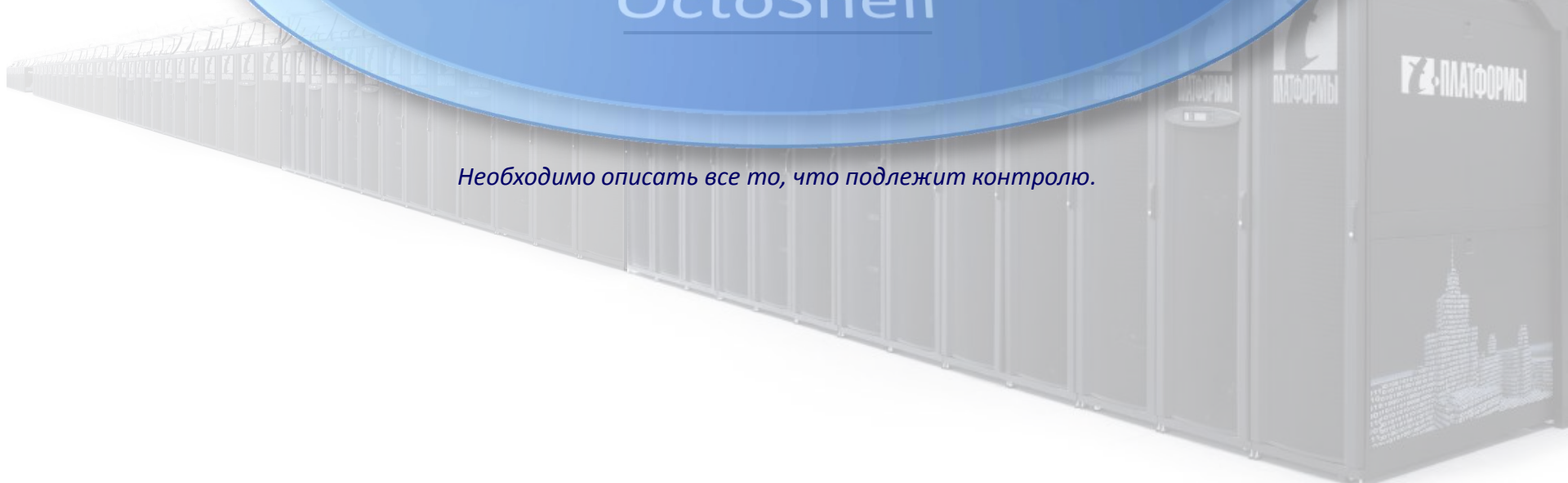
---

*Постоянному мониторингу  
подлежит всё то, что  
влияет на  
эффективность.*



*Гарантия совпадения  
реальности и  
ожиданий о  
работе СКЦ.*

*Необходимо описать все то, что подлежит контролю.*



# Исключительная сложность современных вычислительных систем



# Анализ качества использования СКЦ

---

- Собираем и анализируем очень много разных данных:
  - Данные мониторинга по производительности задач (**DiMMon**)
  - Данные мониторинга по утилизации узлов (без привязки к задачам) (**DiMMon**)
  - Данные от менеджера ресурсов (**Slurm**)
  - Данные от внешних средств анализа (**ML методы**)
  - Данные о найденных проблемах с производительностью (**TASC**)
  - Данные о проектах, организациях, предметных областях (**Octoshell**)
  - Данные о используемых пакетах, компиляторах, библиотеках (**XALT**)
  - Данные о работе файловой системы (**Lustre**)
  - Данные о доступности (ping) и загрузке (LA, disk space) служ. серверов (**Influx**)
  - Данные о ECC ошибках на узлах (**DiMMon**)
  - Данные о числе пользователей и процессов на голове (**Influx**)
  - Данные о корректности работы мониторинга (**TASC**)
  - Данные о доступности (ping) IB коммутаторов (**Influx**)

# TASC: аномальное поведение приложений

## Общая информация

Суперкомпьютер	lomonosov-2
ID задачи	<b>CENSORED</b>
Логин	<b>CENSORED</b>
Статус завершения задачи	COMPLETED
Раздел суперкомпьютера	compute
Число ядер	336
Число узлов	24
Постановка в очередь	10/11/18 18:02:15
Начало счета	10/11/18 19:35:17
Конец счета	10/12/18 05:20:15
Время счета (часы)	9.7

## Производительность

Метрика	Значение	Общая оценка
Средняя загрузка ЦПУ (%)	98.57	good
Среднее LoadAVG	1819.40	low
Среднее IPC	1.15	good
Средняя загрузка ГПУ (%)	0.00	low
Интенсивность передачи данных по MPI (МБ/с)	0.00	low
Интенсивность чтения из файловой системы (МБ/с)	0.00	low
Интенсивность записи в файловую систему (МБ/с)	0.00	low

## Базовые свойства

В задаче мало MPI коммуникаций
Слишком низкая загрузка процессора при данном уровне loadavg
Задача активно работает с памятью
Задача имеет низкую локальность обращений в память

**КРАЙНЕ МНОГО!**

Строка запуска **CENSORED CENSORED CENSORED CENSORED CENSORED**

## Найденные потенциальные проблемы с эффективностью

В данном разделе приведен список проблем с эффективностью, которые были найдены для данной задачи. Для каждой проблемы приведено ее описание (какой признак возникновения проблемы был обнаружен), предположение (в чем, на наш взгляд, может заключаться причина возникновения проблемы) и рекомендация (что мы советуем сделать для ее устранения). Во многих случаях в рекомендации указано, какой тип дальнейшего анализа стоит проводить; реализацию этого функционала планируется выполнить в будущем.

Возможность оценки корректности и/или изменения обнаруженных проблем с эффективностью также планируется добавить в ближайшем будущем. Сейчас эта возможность реализована только на общей странице со списком задач.

Тип	Описание	Предположение	Рекомендация
	Задача запущена в разделе для GPU задач, однако практически не использует графические процессоры.	Неправильно выбран раздел для задачи.	Рекомендуется сменить раздел.
	Задача выполняется аномально неэффективно.	Задача работает некорректно или зависла.	Рекомендуется проверить корректность запуска и при необходимости отменить его.

# TASC: тонкий анализ поведения приложений

- Два идентичных запуска одинаковых приложений.
- Разница во времени исполнения – 2 часа (10%).
- **Истинная причина:** различная сетевая локальность (3 и 6 коммутаторов в запусках)

Найденные проблемы	ID задачи	Начало счета	Конец счета	Статус	Число узлов	Время счета, в часах	Размер задачи (ЦПУ-часы)	Загрузка ЦПУ	Загрузка ГПУ	Load average	IPC	Получено байт по MPI, в МБ	
⋮	⚡	CENSORED	2018-10-20 07:37:52	2018-10-21 11:46:24	completed	24	28.1	9455.8	47.1	0.0	14.0	1.68	550.0
⋮	↔	CENSORED	2018-10-20 11:11:53	2018-10-21 17:18:01	completed	24	30.1	10114.3	46.9	0.0	14.0	1.71	514.2

ⓧ

🔇 🗑️ 👍 ↔ Задача активно работает с MPI сетью, но сетевая локальность плохая (узлы СК расположены далеко друг от друга).

🔇 🗑️ 👍 ⚡ Задача запущена в разделе для GPU задач, однако практически не использует графические процессоры.

🔇 🗑️ 👍 ⚡ Слишком маленькие средние размеры ФС IB пакетов при достаточно высокой интенсивности использования коммуникационной сети.



# TASC: тонкий анализ связки пользователь/пакет

- Данные по одному из пользователей пакета Gromacs:
  - Загрузка процессоров очень высока
  - Сеть, память также неплохо используются
  - НО: графические ускорители вообще не задействованы

name	value	ranking
CPU load	88.6	№ 3 / 20
GPU load	0.0	№ 19 / 20
Load average	24.9	№ 3 / 20
IPC (instructions per second)	0.8	№ 17 / 20
Intensity of FS usage (bytes sent/received per second)	9.2k	№ 16 / 20
Intensity of memory usage (load/store operations per second)	764.1m	№ 8 / 20
Intensity of MPI usage (bytes received per second)	339.4m	№ 11 / 20
Amount of L1 cache misses per second	26.5m	№ 3 / 20
Amount of LLC cache misses per second	84.3k	№ 16 / 20

Хорошо

???

Неплохо

- Результат анализа: своя сборка Gromacs, драйвер GPU не поддерживает последнюю версию CUDA, пользователь был не в курсе...
- После анализа – средняя загрузка GPU равна 43%.

# TASC: проблемы с настройкой пакета (NAMD)

1. На СК Ломоносов-2 есть раздел "pascal" для GPU-задач

2. Обнаружен пользователь со 160 запусками и только 7% средней загрузкой GPU

3. Почти все запуски – это NAMD, причем загрузка GPU у других пакетов, запущенных этим пользователем, в разы выше

4. Все запуски NAMD в "pascal" данным пользователем показывают менее 10% загрузки GPU

5. Этот пользователь – единственный, кто запускает NAMD в "pascal"

6. Многие другие пакеты в "pascal" показывают загрузку GPU гораздо выше (Gromacs – 57%, LAMMPS – 34%)

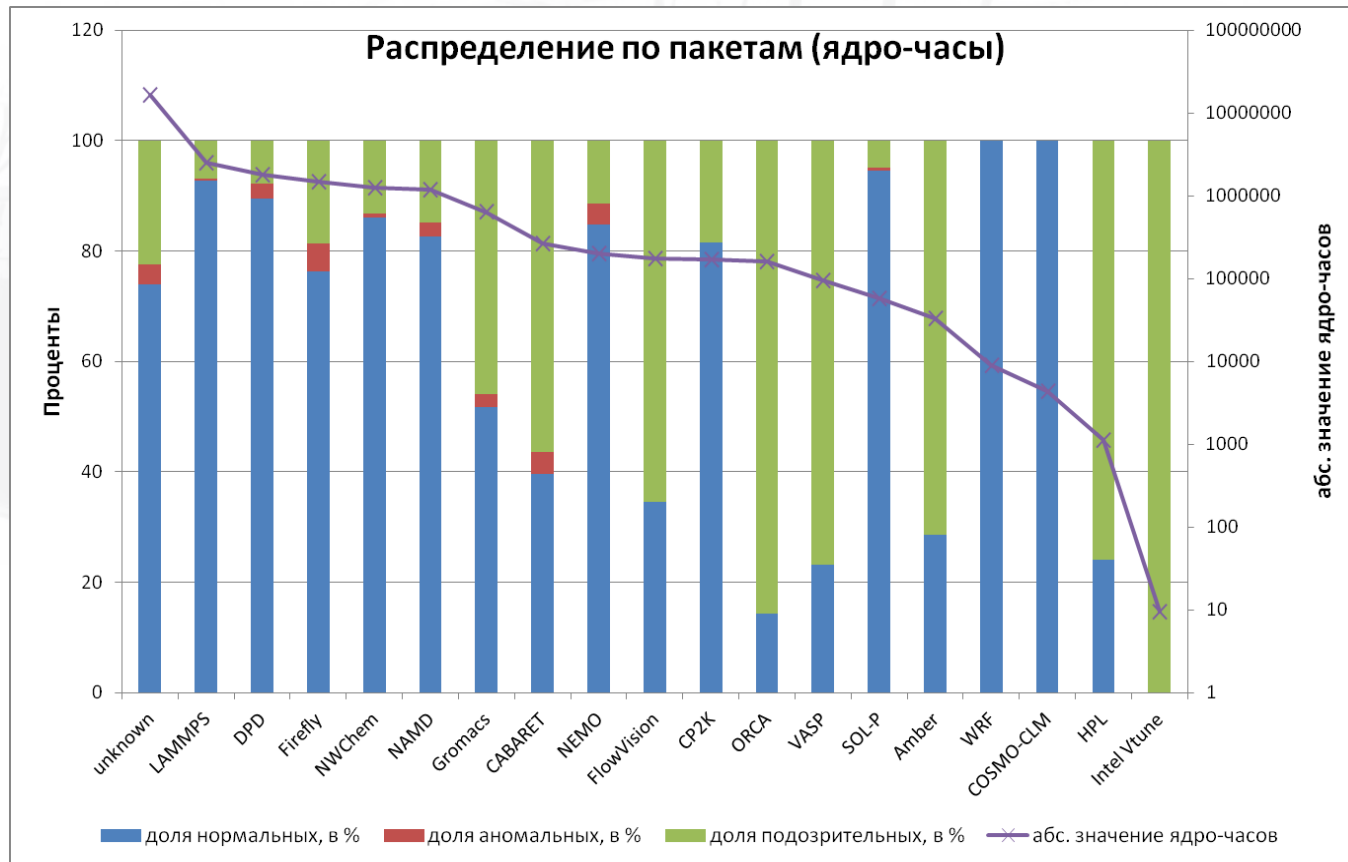
7. Загрузка GPU при запусках NAMD данным пользователем в основном разделе – 57%!

8. Проблема со сборкой NAMD для pascal?

9. Пакет не был адаптирован под новые GPU. Установлена более новая версия NAMD

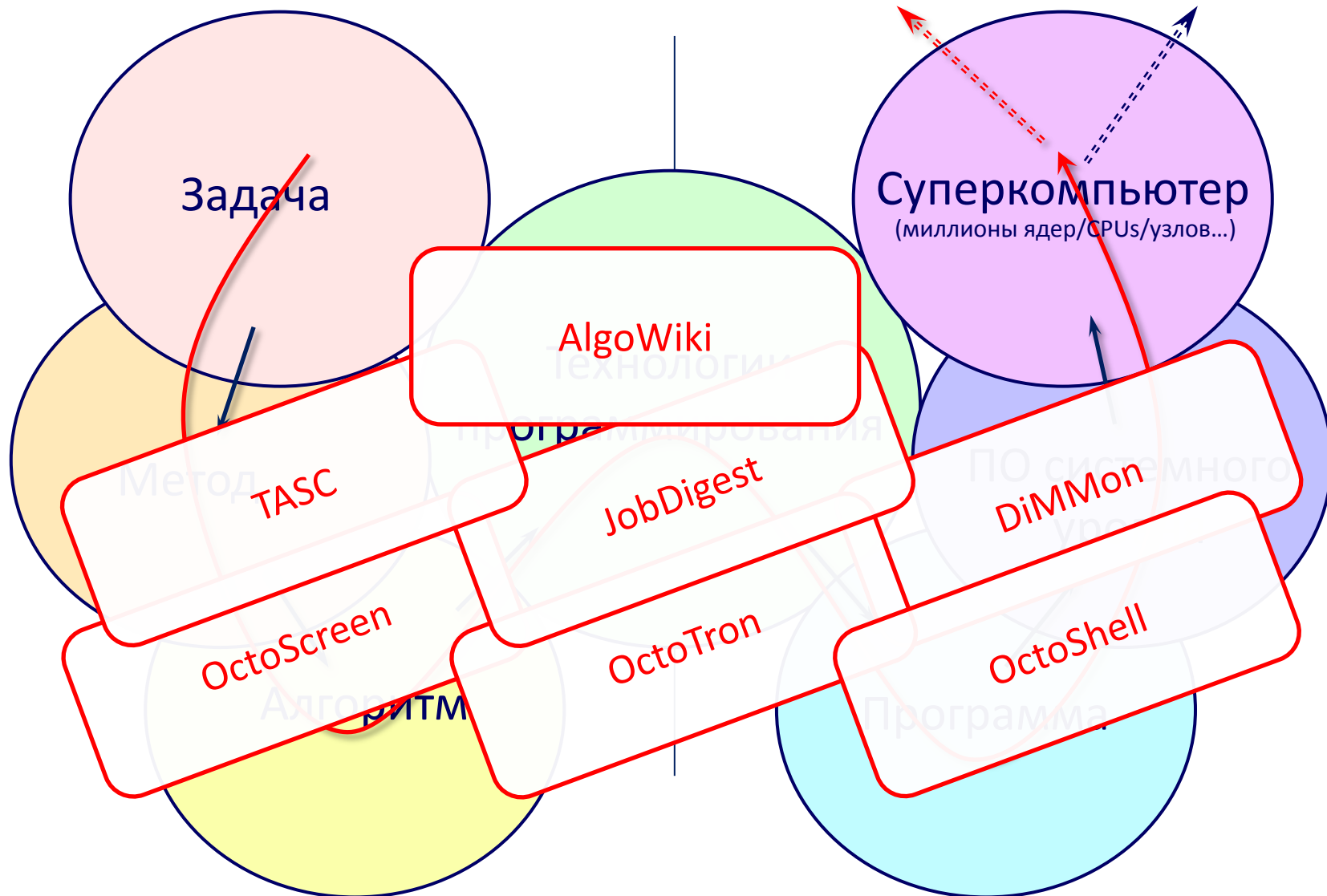
# TASC: статистика по используемым пакетам (На основе пакета XALT)

Ломоносов-2, с 9 октября по 20 декабря 2018:



# Решение задач и вычислительные системы

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

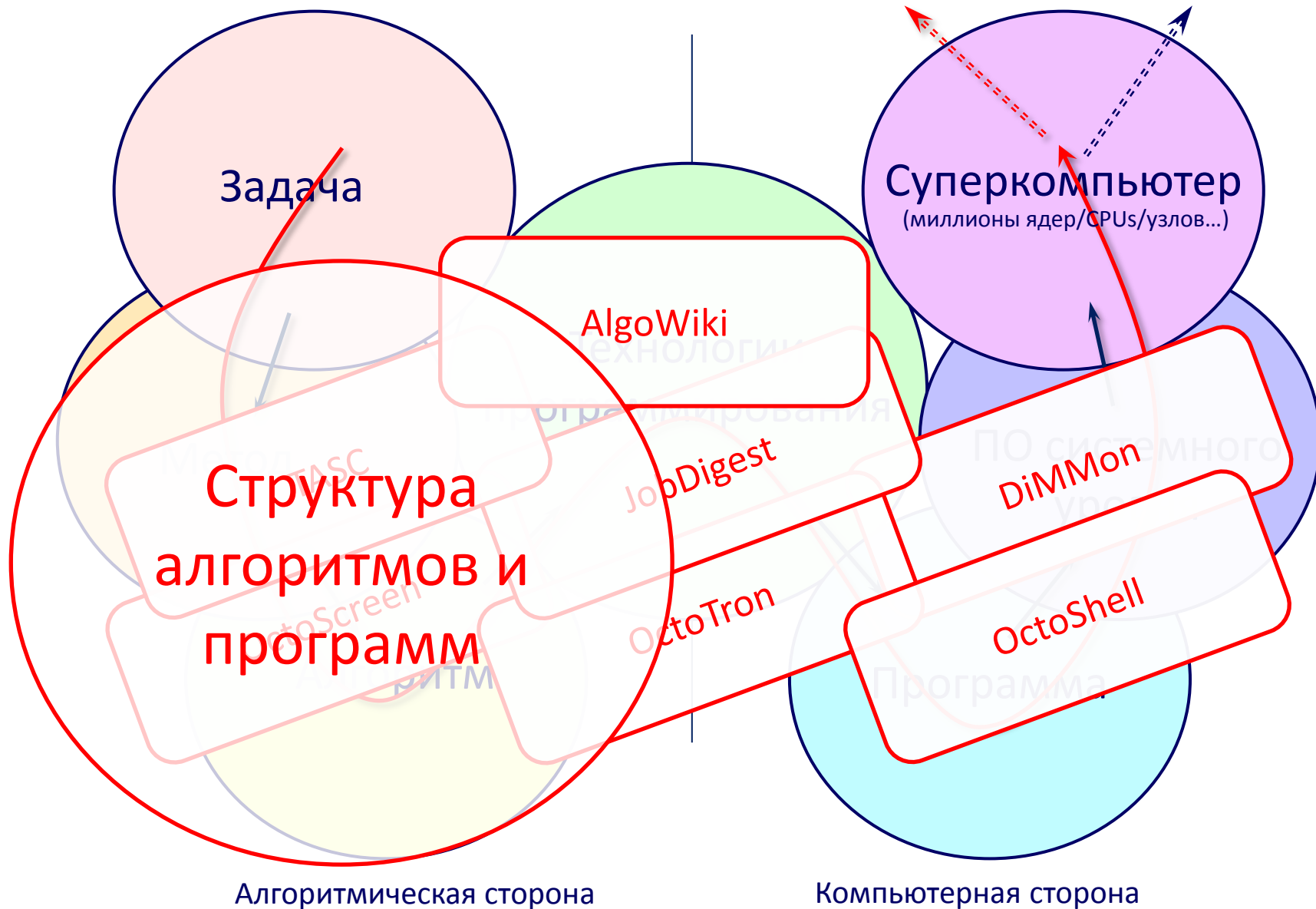


Алгоритмическая сторона

Компьютерная сторона

# Решение задач и вычислительные системы

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



# Ландшафт доступных компьютерных архитектур (Насколько разнообразен компьютерный мир сегодня?)



*И мы, безусловно, должны думать о правильном подборе архитектуры...*

# Вычислительные платформы МГУ



5,5 Pflap/s



1,7 Pflap/s

Технологическая основа:  
Intel Xeon 4/6/10/12... ядер  
SMP-узлы

Intel Xeon Phi (KNL)

NVIDIA 2070 / 2090 / K40 / P100 / V100

NVIDIA DGX-2 (V100)

IBM Power 8 / IBM Blue Gene/P

NEC SX-Aurora Tsubasa

Памяти на узел: от 12GB до 2TB

Разнообразие велико,  
параллелизм – везде...

# *Поколения архитектур и парадигмы программирования (или как часто мы были вынуждены полностью переписывать приложения?)*

*С 1976 года до наших дней:*

*70-е – Векторизация циклов*

*80-е – Распараллеливание циклов (внешних) + Векторизация (внутренних)*

*90-е - MPI*

*середина 90-х - OpenMP*

*середина 2000-х - MPI+OpenMP*

*2010-е - CUDA, OpenCL, MPI+OpenMP+ускорители (GPU, Xeon Phi)*

*...*

***Изменения в архитектуре компьютеров не меняют алгоритмов! Но:***

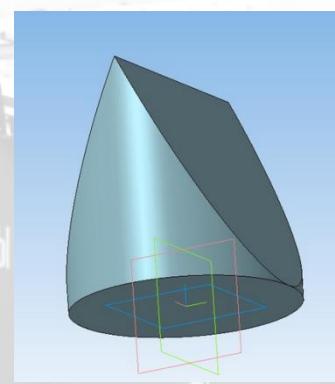
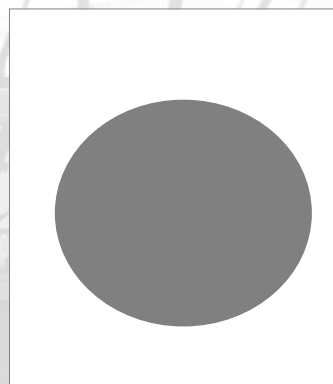
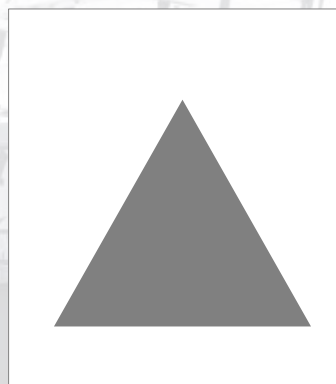
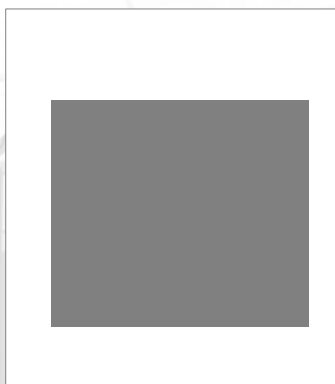
*Для каждого поколения компьютеров мы вынуждены:*

- Анализировать алгоритмы, чтобы понять, как их приспособить под новую компьютерную платформу ;*
- Описывать найденные свойства, чтобы получить эффективную реализацию для новой платформы.*

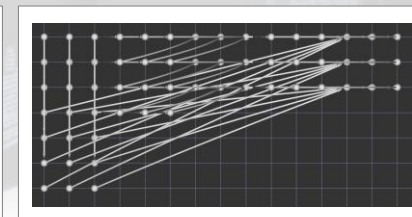
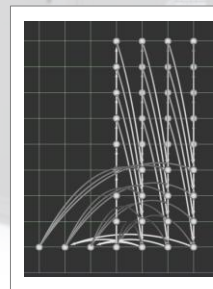
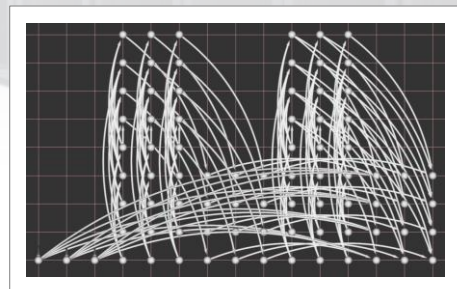
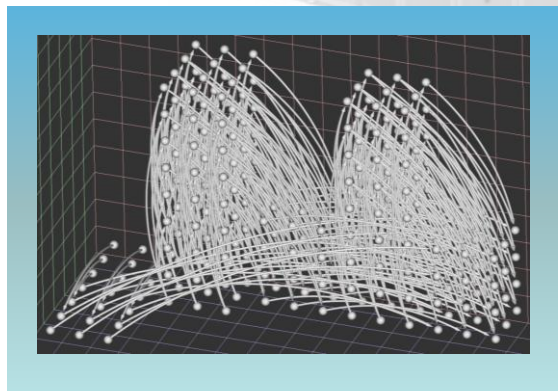


# Изменения в архитектуре компьютеров не меняют алгоритмов!

Эти фигуры разные?



Каковы способы представления этого алгоритма?



...

# Поколения архитектур и парадигмы программирования (или как часто мы были вынуждены полностью переписывать приложения?)

С 1976 года до наших дней:

70-е – Векторизация циклов

80-е – Распараллеливание циклов (внешних) + Векторизация (внутренних)

90-е - MPI

середина 90-х - OpenMP

середина 2000-х - MPI+OpenMP

2010-е - CUDA, OpenCL, MPI+OpenMP+ускорители (GPU, Xeon Phi)

...

Можно ли  
выполнить  
такой анализ  
“раз и навсегда” ?

**Изменения в архитектуре компьютеров не меняют алгоритмов! Но:**

Для каждого поколения компьютеров мы вынуждены:

- Анализировать алгоритмы, чтобы понять, как их приспособить под новую компьютерную платформу ;
- Описывать найденные свойства, чтобы получить эффективную реализацию для новой платформы.

*Что значит “выполнить анализ алгоритма”?*

*Что мы должны найти в алгоритмах?*

*“...выполнить анализ раз и навсегда...” – как записать результаты?*

*Что представляет “единое” / “универсальное” описание алгоритмов?*

*Какие свойства алгоритмов нужно исследовать и описать чтобы получать эффективные реализации в будущем для будущих платформ?*

*Слишком много “простых” вопросов...*

# Структура и свойства алгоритмов

Да, можно: проект AlgoWiki

<http://AlgoWiki-Project.org/>

Можно ли  
выполнить  
такой анализ  
“раз и навсегда” ?

- Анализировать алгоритмы, чтобы понять, как их приспособить под новую компьютерную платформу ;

# Проект AlgoWiki – описание структуры алгоритмов

Файл Правка Вид Журнал Закладки Инструменты Справка

Алговики

algowiki-project.org/ru/Открытая\_энциклопедия\_своих\_свойств\_алгоритмов

Войти Запрос учётной записи

Читать Просмотр История Поиск

## Открытая энциклопедия свойств алгоритмов

**Добро пожаловать! Присоединяйтесь!**

**AlgoWiki** - это открытая энциклопедия по **свойствам алгоритмов и особенностям их реализации** на различных программно-аппаратных платформах от мобильных платформ до экзафлопсных суперкомпьютерных систем с возможностью коллективной работы всего мирового вычислительного сообщества.

Цель **AlgoWiki** - дать исчерпывающее описание алгоритма, которое поможет оценить его потенциал применительно к конкретной параллельной вычислительной платформе. Кроме классических свойств алгоритмов, например, **последовательной сложности**, в AlgoWiki представлены дополнительные сведения, составляющие в совокупности полную картину об алгоритме: **параллельная сложность**, **параллельная структура**, **детерминированность**, **оценки локальности данных**, **эффективность** и **масштабируемость**, **коммуникационный профиль** конкретных реализаций и многие другие.

Читать подробнее: [О проекте](#)

**Структура проекта**

**Классификация алгоритмов** - основной раздел AlgoWiki, содержащий описания всех алгоритмов. Алгоритмы добавляются в подходящий раздел классификации, при необходимости классификация расширяется за счет новых разделов.

**Образцовая статья**

### Разложение Холецкого (метод квадратного корня)

1 Свойства и структура алгоритма

1.1 Общее описание алгоритма

**Свойства алгоритма:**

- Последовательная сложность алгоритма:  $O(n^3)$

**Изображение дня**

**Matrix multiplication performance**

Производительность умножения плотных матриц


**Организация работы**

- **Структура описания свойств алгоритмов**
- Руководство по заполнению разделов описания
- Готовность статей
- Результаты прогона алгоритмов
- Глоссарий
- Помощь

**Участники проекта**

# Структура и свойства алгоритмов

(от мобильных платформ до экзафлопсных суперкомпьютеров)



Информационный граф    Детерминированность  
Вычислительное ядро    Макроструктура    Локальность вычислений  
Производительность    Масштабируемость    Локальность данных  
Сложность    Свойства и особенности    Математическое описание  
Коммуникационный профиль    Эффективность  
Ресурс параллелизма    Вычислительная мощность  
Входные / Выходные данные

# Структура и свойства алгоритмов

(от мобильных платформ до экзафлопсных суперкомпьютеров)

Информационный граф    Детерминированность  
Вычислительное ядро    Макроструктура    Локальность вычислений  
Алгоритмы:    Масштабируемость    Локальность данных  
теоретический потенциал    Сложность и особенности    Математическое описание  
(машинно-независимые свойства)    Сложность    Коммуникационный профиль    Эффективность  
Алгоритмы:    Ресурс параллелизма    Вычислительная мощность  
особенности реализации    Входные / Выходные данные

**AlgoWiki**

<http://AlgoWiki-Project.org>

# Описание структуры и свойств алгоритмов (описание по единой схеме)

## 1 Свойства и структура алгоритмов

- 1.1 Общее описание алгоритма
- 1.2 Математическое описание алгоритма
- 1.3 Вычислительное ядро алгоритма
- 1.4 Макроструктура алгоритма
- 1.5 Схема реализации последовательного алгоритма
- 1.6 Последовательная сложность алгоритма
- 1.7 Информационный граф
- 1.8 Ресурс параллелизма алгоритма
- 1.9 Входные и выходные данные алгоритма
- 1.10 Свойства алгоритма

---

## 2 Программная реализация алгоритма

- 2.1 Особенности реализации последовательного алгоритма
- 2.2 Локальность данных и вычислений
- 2.3 Возможные способы и особенности параллельной реализации алгоритма
- 2.4 Масштабируемость алгоритма и его реализации
- 2.5 Динамические характеристики и эффективность реализации алгоритма
- 2.6 Выводы для классов архитектур
- 2.7 Существующие реализации алгоритма

## 3 Литература



# Описание алгоритмов

(какие свойства должны быть включены в описание?)

Общее описание

Кратко о важном

## 1 Свойства и структура алгоритма

### 1.1 Общее описание алгоритма

**Разложение Холецкого** впервые предложено французским офицером и математиком Андре-Луи Холецким в конце Первой Мировой войны, незадолго до его гибели в бою в августе 1918 г. Идея этого разложения была опубликована в 1924 г. его сослуживцем. Потом оно было использовано поляком Т. Банашевичем в 1938 г. В советской математической литературе называется также методом квадратного корня [1-3]; название связано с характерными операциями, отсутствующими в родственном разложении Гаусса.

Первоначально разложение Холецкого использовалось

Математическое описание

Input data: a symmetric positive definite matrix  $A$  whose elements are real numbers.

Output data: the lower triangular matrix  $L$ , whose elements are real numbers.

The Cholesky algorithm can be represented in the form of the following recurrence relations:

$$l_{11} = \sqrt{a_{11}},$$

$$l_{j1} = \frac{a_{j1}}{l_{11}}, \quad j \in [2, n],$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{p=1}^{i-1} l_{ip}^2}, \quad i \in [2, n].$$

Последовательная сложность

The following number of operations should be performed to compute the Cholesky decomposition of a matrix of order  $n$  using a serial version of the algorithm:

- $n$  square roots,
- $\frac{n(n-1)}{2}$  divisions,
- $\frac{n^3-n}{6}$  multiplications and  $\frac{n^3-n}{6}$  additions (subtractions).

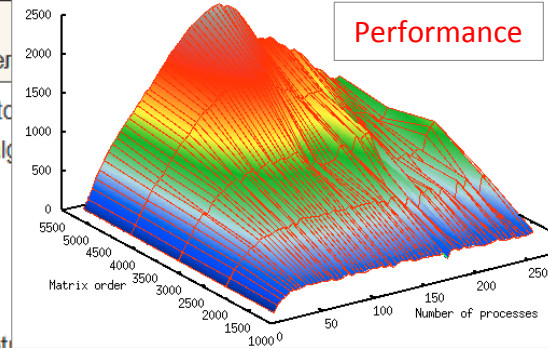
Вычислительное ядро

A computational kernel of its serial version can be composed of the following operations:

$$\sum_{p=1}^{i-1} l_{ip} l_{jp}$$

Свойства алгоритма:

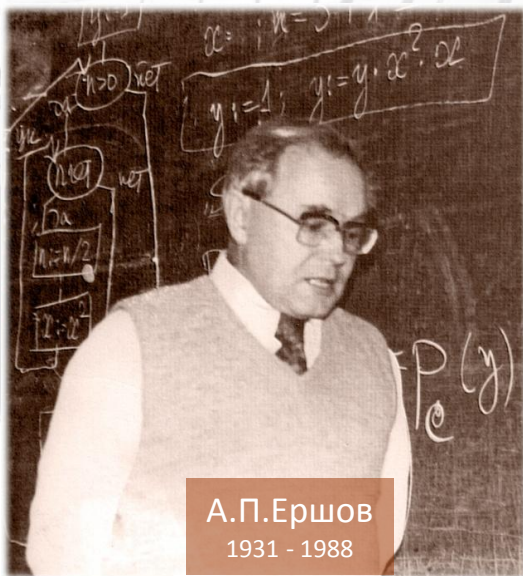
- Последовательная сложность алгоритма:  $O(n^3)$
- Высота ярусно-параллельной формы:  $O(n)$
- Ширина ярусно-параллельной формы:  $O(n^2)$
- Объём входных данных:  $\frac{n(n+1)}{2}$
- Объём выходных данных:  $\frac{n(n+1)}{2}$



Информационная структура



# Информационная структура алгоритмов и программ

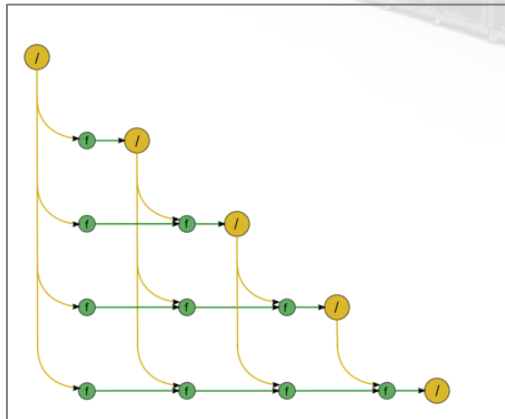
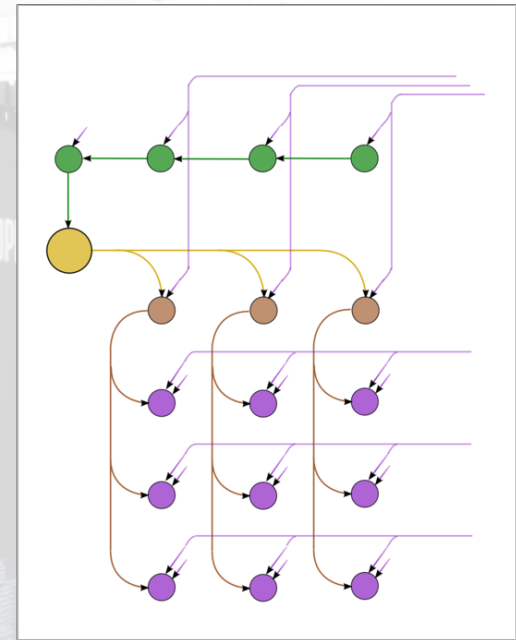
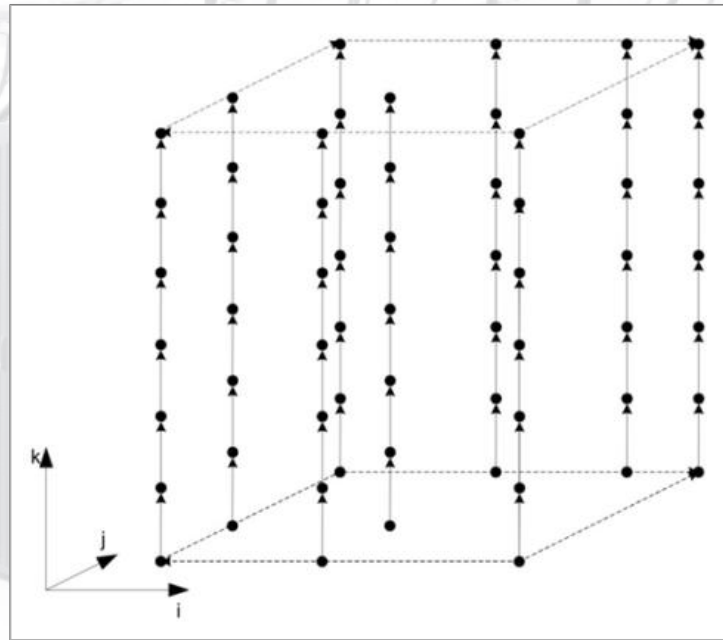
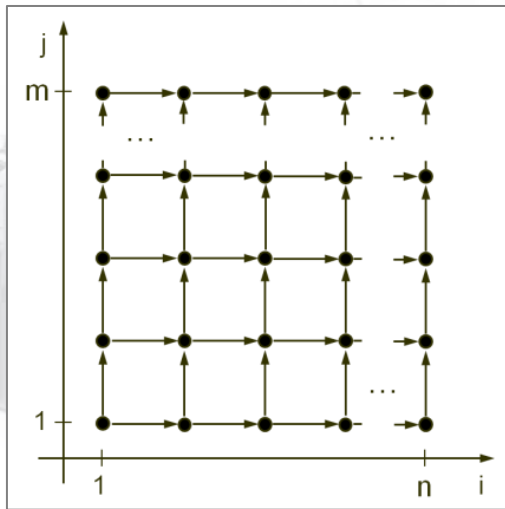


А.П.Ершов  
1931 - 1988



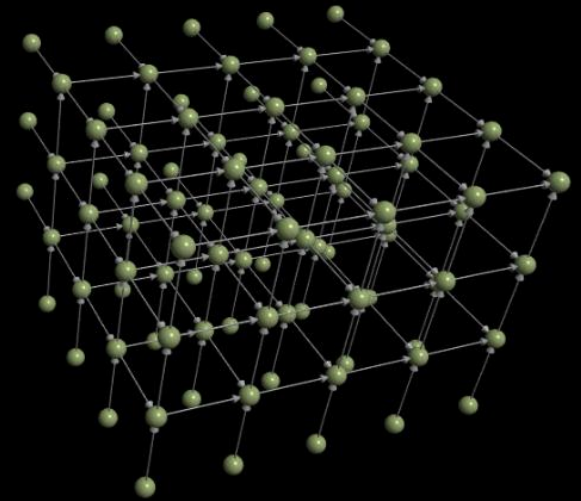
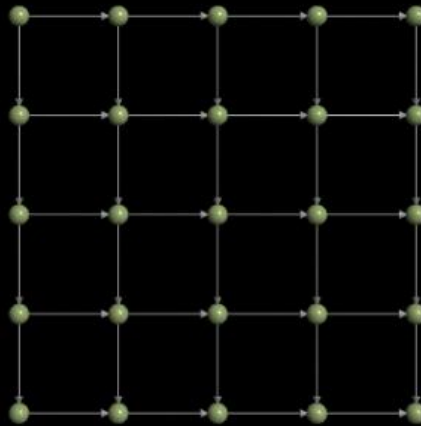
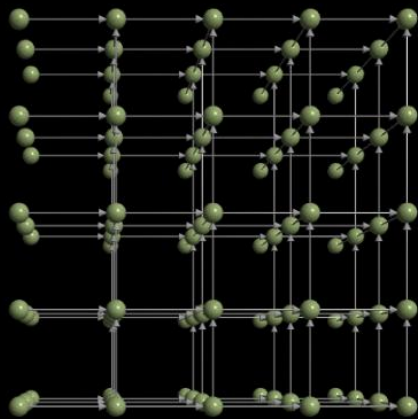
В.В.Воеводин  
1934 - 2007

# Информационная структура: как получать, описывать, показывать... ? (сложности описания алгоритмов)

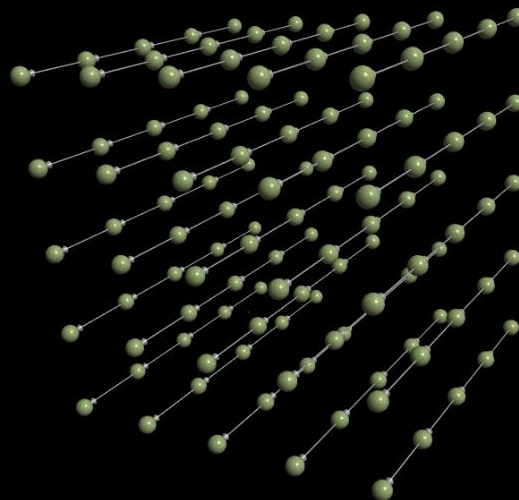
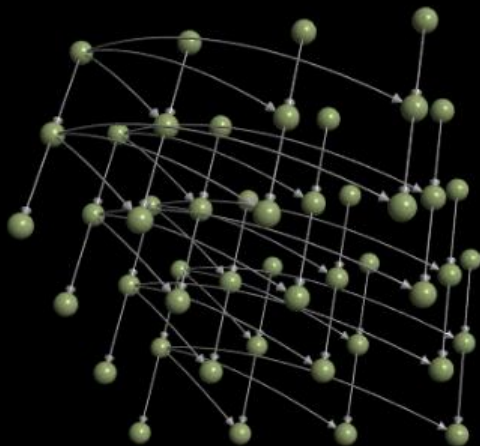


- Как изобразить потенциально бесконечный граф ?
- Как изобразить потенциально многомерный граф ?
- Как показать зависимость структуры графа от размера задачи ?

# Информационная структура алгоритмов и программ



Типовые алгоритмические структуры



“Виден” огромный ресурс параллелизма

# AlgoWiki: классификация алгоритмов

## 1 Linear algebra problems

### 1.1 Matrix and vector operations

#### 1.1.1 Vector operations

- 1. **Dot product**
  - 1. **Dot product**
  - 2. **Fast dot product**
- 2. **Uniform norm of a vector**
- 3. **Dot product**
- 4. **The serial-parallel summation method**

#### 1.1.2 Matrix-vector operations

##### 1.1.2.1 Mapping a nonregular matrix by a vector

- 1. **Dense matrix-vector multiplication**

##### 1.1.2.2 Mapping a matrix of sparsity by a vector

- 1. **Fourier transform**

- 1. **Fast Fourier transform for complex elimination**
  - 1. **Fast Fourier transform for powers of two**
    - 1. **Cody-Tuckey Fast Fourier Transform (radix-2 case)**
    - 2. **Fast Fourier transform for an even number of nodes**
    - 3. **Fast Fourier transform for complex dimension with small prime divisors (1327)**
  - 2. **Fast Fourier transform for prime dimension**

#### 1.1.3 Matrix operations

- 1. **Dense matrix multiplication**
  - 1. **Dense matrix multiplication (serial version for real matrices)**
  - 2. **Strassen's algorithm**

### 1.2 Matrix decompositions

#### 1.2.1 Triangular decomposition

- 1. **Gaussian elimination (finding the LU decomposition)**
  - 1. **LU decomposition using Gaussian elimination without pivoting**
    - 1. **LU decomposition via Gaussian elimination**
    - 2. **Gaussian elimination, comparison scheme for triangular matrices, and its modification**
      - 1. **Comparison scheme for Gaussian elimination Dense matrix**
        - 1. **Gaussian elimination, comparison scheme for triangular matrices, serial version**
        - 2. **Dense doubling algorithm for the LU decomposition of triangular matrices**
        - 3. **Serial-parallel algorithm for the LU decomposition of triangular matrices**
  - 2. **LU decomposition using Gaussian elimination with pivoting**
    - 1. **Gaussian elimination with column pivoting**
    - 2. **Gaussian elimination with row pivoting**
    - 3. **Gaussian elimination with diagonal pivoting**
    - 4. **Gaussian elimination with complete pivoting**
- 2. **Cholesky method**
  - 1. **Cholesky decomposition**

#### 2. **Available singular decompositions for matrices of special form**

##### 2.1 Unitary-orthogonal factorizations

- 1. **QR decomposition of dense nonsingular matrices**
  - 1. **Sivina (rotation) method for the QR decomposition of a matrix**
    - 1. **Sivina method**
  - 2. **Householder (reflector) method for the QR decomposition of a matrix**
    - 1. **Householder (reflector) method for the QR decomposition of a square matrix, real polynomial version**
  - 3. **Orthogonalization method**
    - 1. **Classical orthogonalization method**
    - 2. **Orthogonalization method with reorthogonalization**
    - 3. **Triangular decomposition of Gram matrix**
- 2. **QR decomposition methods for dense Hessenberg matrices**
  - 1. **Sivina (rotation) method for the QR decomposition of a (real) Hessenberg matrix**
  - 2. **Householder (reflector) method for the QR decomposition of a (real) Hessenberg matrix**
    - 1. **Classical polynomial Sivina (rotation) method for reducing a matrix to Hessenberg form**

##### 4. **Reducing matrices to compact form**

- 1. **Unitary reduction to Hessenberg form**
  - 1. **Householder (reflector) method for reducing a matrix to Hessenberg form**
  - 2. **Classical polynomial Householder (reflector) method for reducing a matrix to Hessenberg form**
- 2. **Sivina (rotation) method for reducing a matrix to Hessenberg form**
  - 1. **Classical polynomial Sivina (rotation) method for reducing a matrix to Hessenberg form**
- 3. **Unitary reduction to orthogonal form**
  - 1. **Householder (reflector) method for reducing to orthogonal form**
    - 1. **Householder (reflector) method for reducing a symmetric matrix to orthogonal form**
    - 2. **Householder (reflector) method for reducing a complex Hermitian matrix to a symmetric orthogonal form**
  - 2. **Sivina (rotation) reduction to orthogonal form**
- 2. **QR value decomposition (finding eigenvalues and eigenvectors)**

##### 2. Unitary non-similarity reduction to compact forms

- 1. **Unitary reduction to bidiagonal form**
  - 1. **Householder (reflector) reduction of a matrix to bidiagonal form**
  - 2. **Sivina (rotation) reduction of a matrix to bidiagonal form**
- 2. **Singular value decomposition**
  - 1. **Singular value decomposition (finding singular values and singular vectors)**

### 1.3 Solving systems of linear algebraic equations

#### 1. Direct methods

- 1. **Implicit benchmark**
- 2. **Harries of a special form**
  - 1. **Triangular matrices**
    - 1. **Forward substitution**
    - 2. **Backward substitution**
  - 2. **Bi-diagonal matrices**
    - 1. **Forward and backward substitution for bidiagonal matrices**
    - 2. **Serial-parallel variants of the backward substitution**
    - 3. **Serial-parallel variants of the backward substitution**
- 3. **Methods for solving orthogonal SLSs**
  - 1. **Methods based on the conventional LU decomposition**
    - 1. **Thomas algorithm**
      - 1. **Thomas algorithm, polynomial version**
      - 2. **Revised Thomas algorithm, polynomial version**
    - 2. **Dense doubling algorithm**
      - 1. **Dense doubling algorithm for the LU decomposition of bidiagonal matrices**
      - 2. **Dense doubling algorithm for solving bi-diagonal SLSs**
      - 3. **Serial-parallel variants of solving orthogonal matrices based on the LU decomposition and backward substitutions**
  - 2. **Other methods**
    - 1. **Reduction method**
      - 1. **Complex reduction method**
      - 2. **Reduction method repeated for a new (right-hand side)**
    - 2. **Two-sided Thomas algorithm**
      - 1. **Two-sided Thomas algorithm, polynomial version**
      - 2. **Revised two-sided Thomas algorithm, polynomial version**
    - 3. **Cyclic reduction**
      - 1. **Complex cyclic reduction**
      - 2. **Cyclic reduction repeated for a new (right-hand side)**
    - 4. **Block reduction method**
- 3. **Methods for solving block triangular matrices**
  - 1. **Block forward substitution (real version)**
  - 2. **Block backward substitution (real version)**
  - 3. **Methods for solving block bi-diagonal matrices**
    - 1. **Forward and backward substitution for block bi-diagonal matrices**
    - 2. **Dense doubling algorithm for solving block bi-diagonal matrices**
  - 4. **Serial-parallel variants of the block backward substitution for solving block bi-diagonal matrices**

#### 3. **Solving systems of linear algebraic equations with coefficient matrices of special form whose structure are known**

- 2. **Serial methods for solving systems of linear algebraic equations**
  - 1. **High Performance Conjugate Gradient (HPCG) benchmark**
  - 2. **Block-cyclic gradient method (BCGM)**
  - 3. **Block-cyclic algorithm**

### 1.4 Solving eigenvalue problems

- 1. **QR value decomposition (finding eigenvalues and eigenvectors)**
  - 1. **QR algorithm**
    - 1. **QR algorithm as implemented in SCLAPACK**
      - 1. **Classical polynomial Householder (reflector) method for reducing a matrix to Hessenberg form**
      - 2. **Hessenberg QR algorithm as implemented in SCLAPACK**
    - 2. **Symmetric QR algorithm as implemented in SCLAPACK**
      - 1. **Householder (reflector) method for reducing a symmetric matrix to orthogonal form**
    - 3. **Symmetric orthogonal QR algorithm as implemented in SCLAPACK**
    - 4. **QR algorithm for complex Hermitian matrices as implemented in SCLAPACK**
      - 1. **Householder (reflector) method for reducing complex Hermitian matrix to a symmetric orthogonal form**
      - 2. **Symmetric orthogonal QR algorithm as implemented in SCLAPACK**
  - 2. **The Jacobi (rotation) method for solving the symmetric eigenvalue problem**
    - 1. **Classical Jacobi (rotation) method with pivoting for symmetric matrices**
    - 2. **Serial Jacobi (rotation) method for symmetric matrices**
    - 3. **Serial Jacobi (rotation) method with threshold for symmetric matrices**
  - 3. **Lanczos algorithm**
    - 1. **Lanczos algorithm in exact arithmetic (with reorthogonalization)**
- 2. **Unitary eigenvalue problem**
  - 1. **Singular value decomposition**
- 3. **Implicit value decomposition (finding singular values and singular vectors)**
  - 1. **Jacobi (rotation) method for finding singular values**
    - 1. **Serial Jacobi (rotation) method for finding singular values**
    - 2. **Jacobi method with a sparse choice of rotations for finding singular values**
  - 2. **QR algorithm as applied to singular value decomposition arrays**

### 1.5 Algebra of polynomials

- 1. **Horner's method**

## 2 Algorithms on lists and arrays

### 2.1 Search algorithms

- 1. **Linear search** (Finding an item in an arbitrary list,  $O(n)$ )
- 2. **Binary search** (Finding the position of an element within a sorted array,  $O(\log(n))$ )

### 2.2 Sorting algorithms

- 1. **Bray's tree sort**
- 2. **Bubble sort**
- 3. **Large sort (serial and parallel variants)**

### 2.3 Graph algorithms

- 1. **Graph traversal**
  - 1. **Breadth-first search (BFS)**
  - 2. **Depth-first search (DFS)**
  - 3. **Single Source Shortest Path (SSSP)**
    - 1. **Breadth-first search (BFS) for unweighted graphs**
    - 2. **Dijkstra's algorithm**
    - 3. **Bellman-Ford algorithm**
    - 4. **Shortest path algorithm**
  - 4. **All Pairs Shortest Path (APSP)**
    - 1. **Floyd-Warshall algorithm**
    - 2. **Transitive closure of a directed graph**
      - 1. **Floyd's algorithm**
  - 5. **Longest Shortest Path**
  - 6. **Construction of a minimum spanning tree (MST)**
    - 1. **Kruskal's algorithm**
    - 2. **Prim's algorithm**
    - 3. **Edmonds-Karp algorithm**
- 2. **Search for isomorphic subgraphs**
  - 1. **Vitányi's algorithm**
  - 2. **RT algorithm**
- 3. **Block connectivity**
  - 1. **Block-tripartition algorithm for finding the connected components**
  - 2. **Block partition**
  - 3. **Block-tripartition algorithm for finding the connected components**
  - 4. **Block-tripartition algorithm for finding the connected components**
  - 5. **Block-tripartition algorithm for finding the connected components**
  - 6. **Block-tripartition algorithm for finding the connected components**
  - 7. **Block-tripartition algorithm for finding the connected components**
  - 8. **Block-tripartition algorithm for finding the connected components**
- 4. **Finding maximal flow in a transportation network**
  - 1. **Ford-Fulkerson algorithm**
  - 2. **Edmonds-Karp algorithm**
- 5. **Finding minimal cost flow in a transportation network**
  - 1. **Edmonds-Karp algorithm**
  - 2. **Edmonds-Karp algorithm**
  - 3. **Edmonds-Karp algorithm**
- 6. **Shortest path problem**
  - 1. **Edmonds-Karp algorithm**
  - 2. **Edmonds-Karp algorithm**
  - 3. **Edmonds-Karp algorithm**

## 3 Computational geometry

### 3.1 Finding the convex hull of a point set

- 1. **Convex hull of a point set**
- 2. **Convex hull of a point set**
- 3. **Convex hull of a point set**
- 4. **Convex hull of a point set**
- 5. **Convex hull of a point set**
- 6. **Convex hull of a point set**
- 7. **Convex hull of a point set**

### 3.2 Computer graphics

- 1. **Line drawing algorithm (approximating a line segment discrete graphical model)**
- 2. **Drawing a cube from a three-dimensional scene**
- 3. **Drawing a rectangle from a three-dimensional scene**
- 4. **Global illumination: Ray-tracing of reflection and refraction from other objects**

## 4 Computer analysis and modeling

### 4.1 Computer benchmarks

- 1. **High Performance Conjugate Gradient (HPCG) benchmark**
- 2. **Implicit benchmark**

### 4.2 Algorithms of quantum system simulation

- 1. **Algorithm of quantum computation simulation**
  - 1. **Quantum Fourier transform of a real vector**
  - 2. **Quantum Fourier transform of a real vector**
  - 3. **Quantum Fourier transform simulation**

# *AlgoWiki: Problem – Method – Algorithm – Implementation*

*(подробное описание по всей цепочке)*



Problem



Method



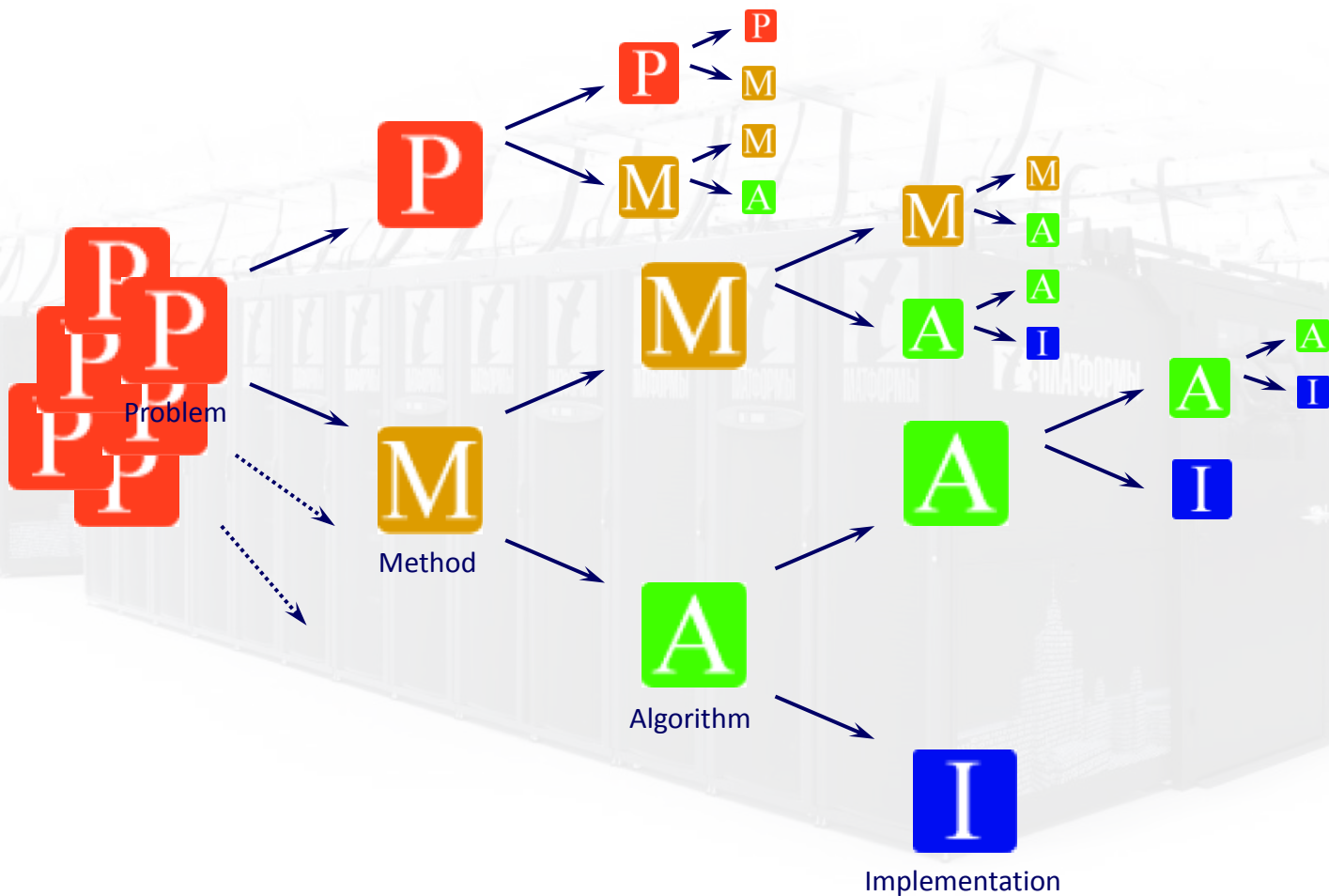
Algorithm



Implementation

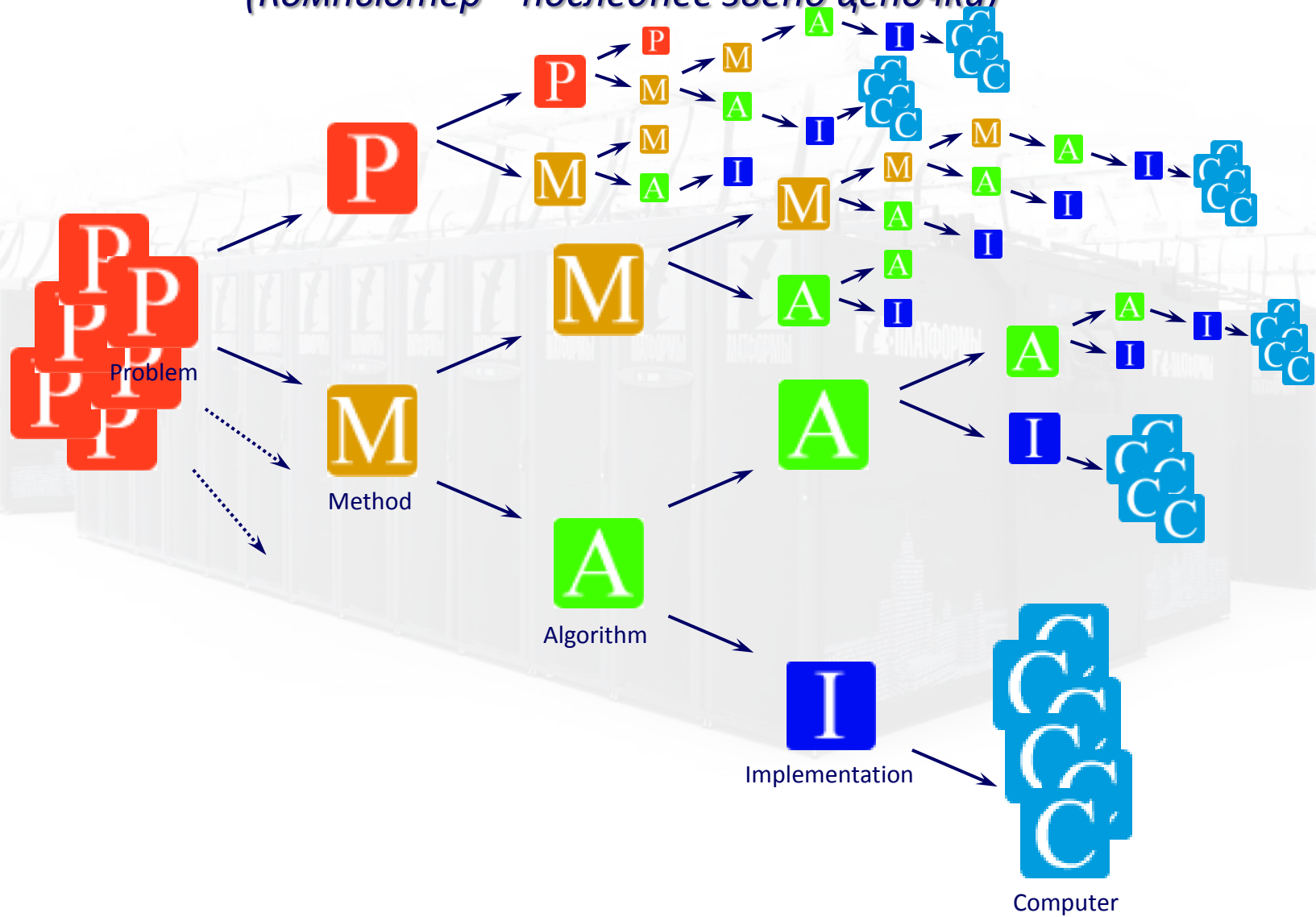
# AlgoWiki: Problem – Method – Algorithm – Implementation

(на практике ситуация намного сложнее и разнообразнее)



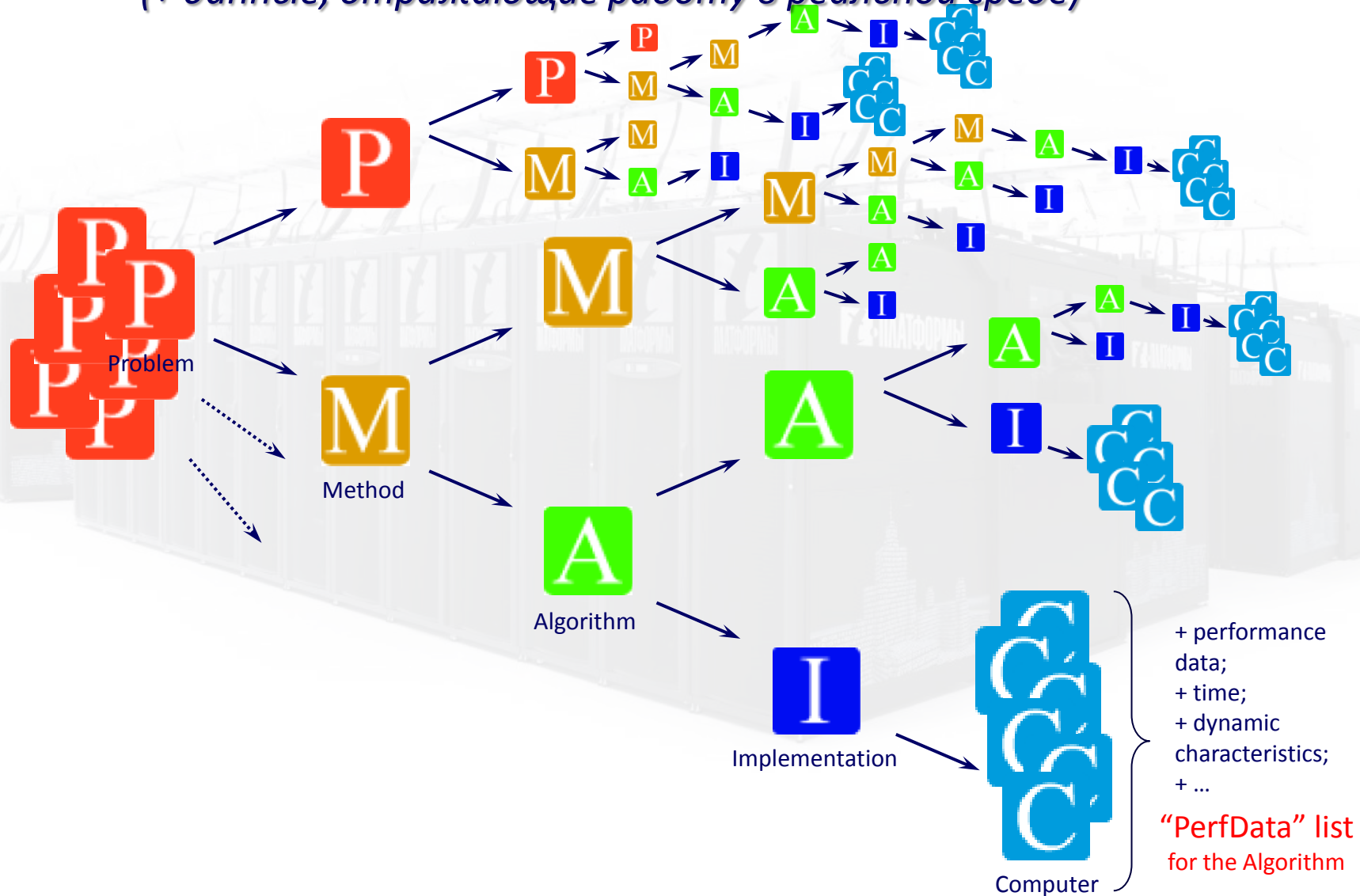
# AlgoWiki: от Задачи к различным Реализациям

(Компьютер – последнее звено цепочки)

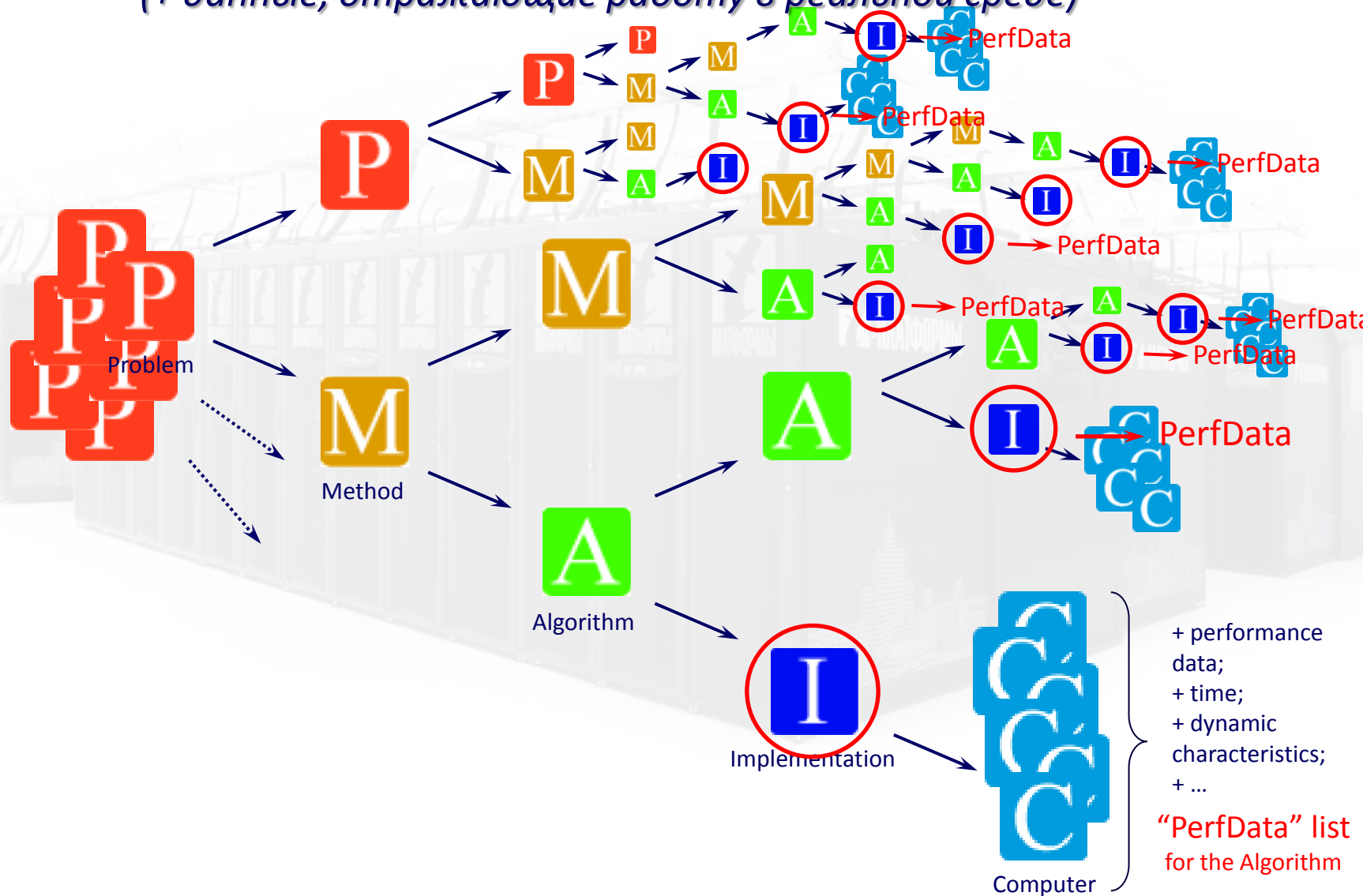




# AlgoWiki: от Задачи к различным Реализациям (+ данные, отражающие работу в реальной среде)



# AlgoWiki: от Задачи к различным Реализациям (+ данные, отражающие работу в реальной среде)

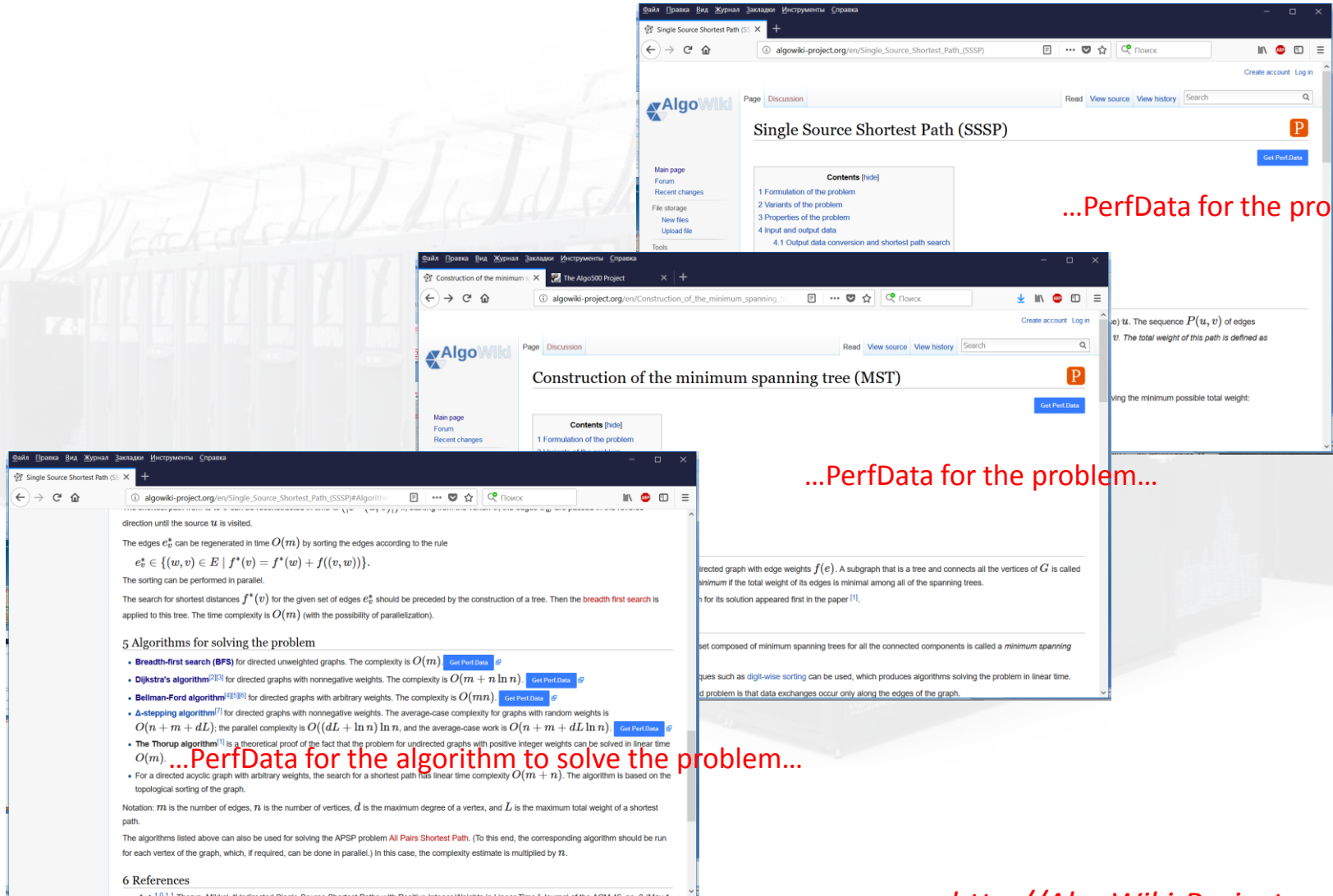


*Как это работает ?*

*Что мы можем получить из AlgoWiki ?*



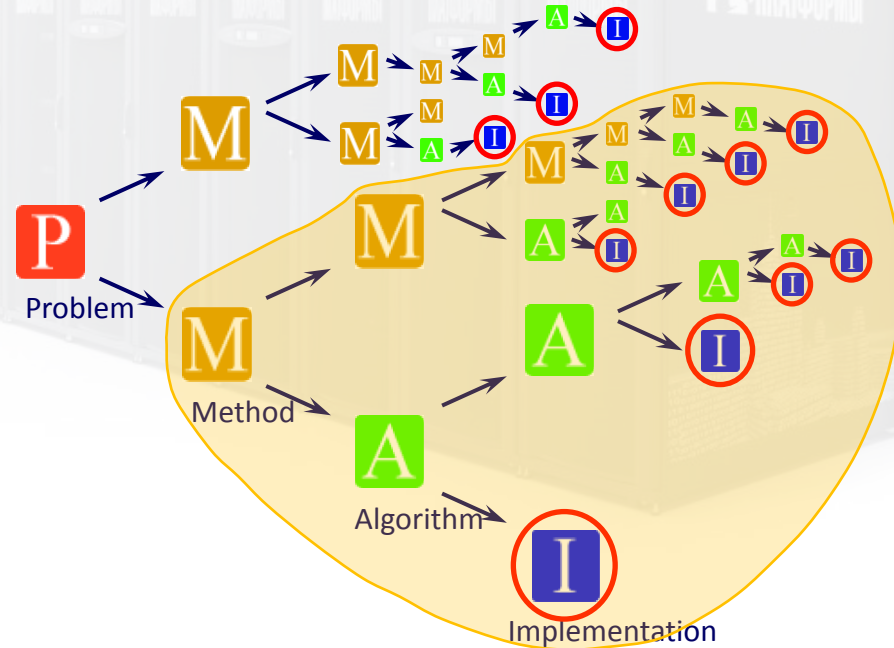
# Извлекаем наборы PerfData из AlgoWiki...



# PerfData для задачи “Strongly Connected Components” (Метод = Forward-Backward)



- сравнить различные алгоритмы, реализации и вычислительные платформы для данной задачи и для данного метода;



# PerfData для задачи “Strongly Connected Components” (Method = Forward-Backward)

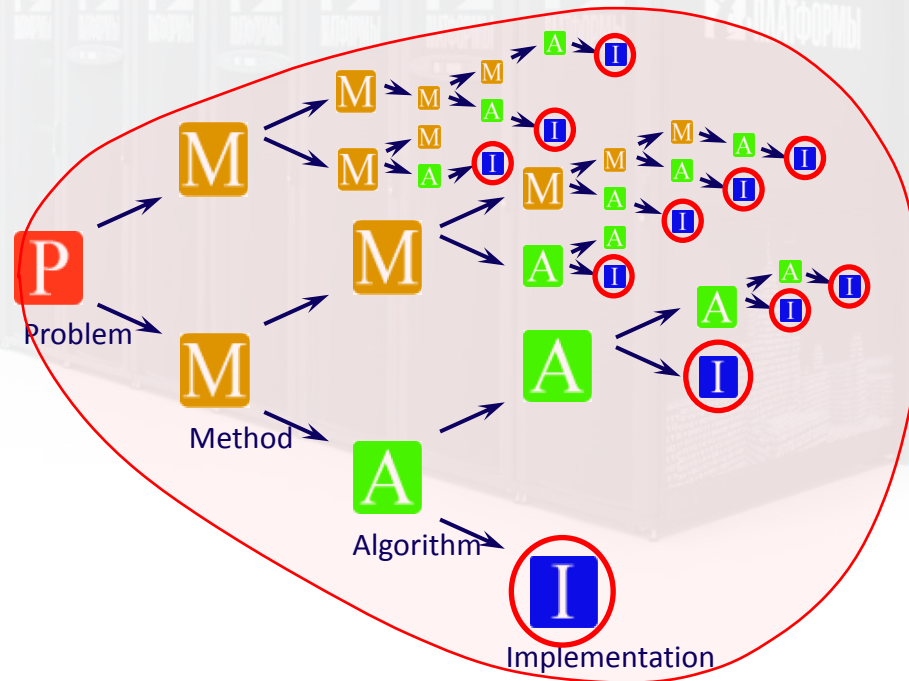
Rating	Method	Implementation	Platform	MTEPS	GraphType	GraphSize
1	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	634,00	RMAT	2^20
2	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	620,00	RMAT	2^21
3	Forward-Backward	RCC for CPU	Lomonosov-2	564,00	RMAT	2^24
4	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	544,00	RMAT	2^22
5	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	528,00	RMAT	2^23
6	Forward-Backward	RCC for CPU	Lomonosov-2	498,00	RMAT	2^26
7	Forward-Backward	RCC for CPU	Lomonosov-2	497,00	RMAT	2^25
8	Forward-Backward	RCC for CPU	Lomonosov-2	486,00	RMAT	2^27
9	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	456,00	RMAT	2^25
10	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	453,00	RMAT	2^24
11	Forward-Backward	RCC for CPU	Lomonosov-2	452,00	RMAT	2^22
12	Forward-Backward	RCC for CPU	Lomonosov-2	440,24	SSCA-2	2^21
13	Forward-Backward	RCC for CPU	Lomonosov-2	432,00	RMAT	2^23
14	Forward-Backward	RCC for CPU	Lomonosov-2	426,00	RMAT	2^21
15	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	426,00	RMAT	2^26
16	Forward-Backward	RCC for CPU	Lomonosov-2	418,00	RMAT	2^20
17	Forward-Backward	PBGL MPI	IBM BlueGene/P	232,86	RMAT	2^20
18	Forward-Backward	RCC for GPU	Lomonosov-2	195,00	RMAT	2^18
19	Forward-Backward	PBGL MPI	Lomonosov	91,07	RMAT	2^21
20	Forward-Backward	RCC for CPU	Lomonosov-2	55,44	RMAT	2^18
21	Forward-Backward	RCC for CPU	IBM Regatta	53,60	SSCA-2	2^18
22	Forward-Backward	PBGL MPI	IBM BlueGene/P	45,75	RMAT	2^20
23	Forward-Backward	RCC for GPU	Lomonosov	44,78	RMAT	2^16
24	Forward-Backward	RCC for CPU	Lomonosov	42,00	RMAT	2^22
25	Forward-Backward	RCC for CPU	Lomonosov	41,00	RMAT	2^20
26	Forward-Backward	RCC for CPU	IBM Regatta	36,90	RMAT	2^18
27	Forward-Backward	RCC for CPU	Lomonosov	32,54	RMAT	2^20
28	Forward-Backward	PBGL MPI	IBM BlueGene/P	13,39	SSCA-2	2^16
29	Forward-Backward	PBGL MPI	IBM BlueGene/P	13,12	SSCA-2	2^18
30	Forward-Backward	RCC for CPU	Lomonosov	10,05	SSCA-2	2^20
31	Forward-Backward	RCC for CPU	Lomonosov	9,20	SSCA-2	2^18
32	Forward-Backward	RCC for CPU	Lomonosov	8,30	SSCA-2	2^20

# PerfData для задачи “*Strongly Connected Components*”

(различные методы, алгоритмы, реализации, компьютеры)



- Необходимо сравнить различные подходы к решению данной задачи;



# PerfData для задачи “Strongly Connected Components” (различные методы, алгоритмы, реализации, компьютеры)

Rating	Method	Implementation	Platform	MTEPS	GraphType	GraphSize
1	Shiloach-Vishkin	Ligra	Lomonosov-2	1307,00	RMAT	2^26
2	Shiloach-Vishkin	Ligra	Lomonosov-2	986,00	RMAT	2^23
3	Shiloach-Vishkin	Ligra	Lomonosov-2	947,00	RMAT	2^22
4	Shiloach-Vishkin	Ligra	Lomonosov-2	894,00	RMAT	2^24
5	Shiloach-Vishkin	Ligra	Lomonosov-2	864,00	RMAT	2^25
6	Shiloach-Vishkin	Ligra	Lomonosov-2	830,00	RMAT	2^20
7	Shiloach-Vishkin	Ligra	Lomonosov-2	782,00	RMAT	2^21
8	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	634,00	RMAT	2^20
9	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	620,00	RMAT	2^21
10	Forward-Backward	RCC for CPU	Lomonosov-2	564,00	RMAT	2^24
11	Shiloach-Vishkin	GAP	Lomonosov-2	547,00	RMAT	2^20
12	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	544,00	RMAT	2^22
13	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	528,00	RMAT	2^23
14	Forward-Backward	RCC for CPU	Lomonosov-2	498,00	RMAT	2^26
15	Forward-Backward	RCC for CPU	Lomonosov-2	497,00	RMAT	2^25
16	Forward-Backward	RCC for CPU	Lomonosov-2	486,00	RMAT	2^27
17	Shiloach-Vishkin	GAP	Lomonosov-2	480,00	RMAT	2^22
18	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	456,00	RMAT	2^25
19	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	453,00	RMAT	2^24
20	Forward-Backward	RCC for CPU	Lomonosov-2	452,00	RMAT	2^22
21	Forward-Backward	RCC for CPU	Lomonosov-2	440,24	SSCA-2	2^21
22	Forward-Backward	RCC for CPU	Lomonosov-2	432,00	RMAT	2^23
23	Forward-Backward	RCC for CPU	Lomonosov-2	426,00	RMAT	2^21
24	Forward-Backward	RCC for GPU	Lomonosov-2 (P100)	426,00	RMAT	2^26
25	Forward-Backward	RCC for CPU	Lomonosov-2	418,00	RMAT	2^20
26	Shiloach-Vishkin	GAP	Lomonosov-2	387,00	RMAT	2^23
27	Shiloach-Vishkin	GAP	Lomonosov-2	335,00	RMAT	2^21
28	Forward-Backward	PBGL MPI	IBM BlueGene/P	232,86	RMAT	2^20
29	Shiloach-Vishkin	GAP	Lomonosov-2	231,00	RMAT	2^24
30	Forward-Backward	RCC for GPU	Lomonosov-2	195,00	RMAT	2^18
31	Shiloach-Vishkin	GAP	Lomonosov-2	180,00	RMAT	2^25
32	Shiloach-Vishkin	GAP	Lomonosov-2	174,00	RMAT	2^26

Данные в репозитории AlgoWiki могут представляться разными людьми



# Что означает “представить набор PerfData в AlgoWiki”?

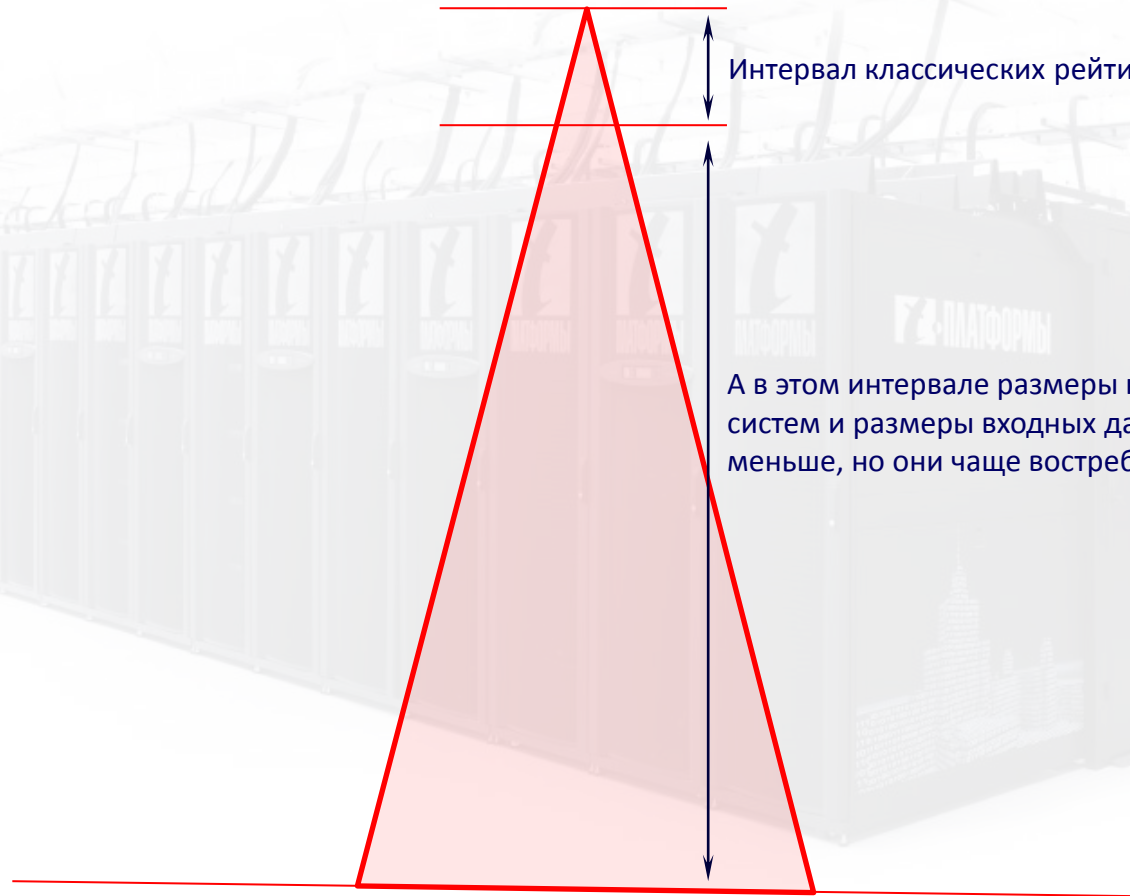
Path to the algorithm in AlgoWiki 1	Implementation details 2	Computer platform 3	Input data 4	Performance data 5
-------------------------------------	--------------------------	---------------------	--------------	--------------------

1. Информация о всей цепочке: **P** → **M** → **A** → **I** → **C** (путь в AlgoWiki)
2. Исходный код или ссылка на пакет. Компилятор, опции и флаги компиляции, использованные библиотеки, make-файлы, параметры запуска...
3. Конфигурация вычислительной системы со всеми необходимыми деталями.  
- конфигурации разного размера: Max / Medium / Small.
4. Описание входных данных (matrices (dense, sparse, symmetric, ...), graphs (RMAT, SSCA-2...), sizes, ...)  
- входные данные разного размера: Max / Medium / Small.  
- параметры запуска (block size, processor matrix...)
5. Performance data:  
- Время, Flops, MTEPS, Gflops/Watt, эффективность ( $R_{\max}/R_{\text{peak}}$ ), ...

**Важно: нам нужно хранить все данные и детали, чтобы иметь возможность воспроизводить представленные в AlgoWiki данные по производительности.**

# *Max / Medium / Small – что это значит?*

*(конфигурация компьютера, размер входные данные)*

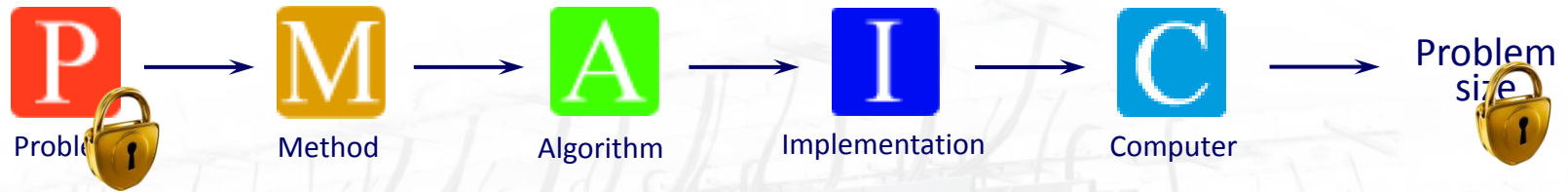


Интервал классических рейтингов “Top/Graph/HPCG-500”

А в этом интервале размеры конфигураций вычислительных систем и размеры входных данных не рекордные, они меньше, но они чаще востребованные на практике.

# PerfData для задачи “Single Source Shortest Path”

(Как быстрее решить задачу данного размера?)



Method	Implementation	Computing Platform	MTEPS	GraphType	GraphSize
Bellman-Ford	RCC for GPU	Lomonosov	1309,0	SSCA-2	2 <sup>20</sup>
Delta-Stepping	GAP	Lomonosov-2	512,0	RMAT	2 <sup>20</sup>
Bellman-Ford	Ligra	Lomonosov-2	511,0	RMAT	2 <sup>20</sup>
Bellman-Ford	RCC for GPU	Lomonosov	452,9	SSCA-2	2 <sup>20</sup>
Bellman-Ford	RCC for CPU	Lomonosov-2	418,0	RMAT	2 <sup>20</sup>
Bellman-Ford	Graph500 MPI	Lomonosov	350,0	RMAT	2 <sup>20</sup>
Bellman-Ford	RCC for CPU	Lomonosov	204,1	RMAT	2 <sup>20</sup>
Bellman-Ford	RCC for CPU	Lomonosov	183,5	SSCA-2	2 <sup>20</sup>
Dijkstra's	PBGL MPI	Cluster / "Angara" interconnect	150,0	SSCA-2	2 <sup>20</sup>
Bellman-Ford	Graph500 MPI	Lomonosov	120,0	RMAT	2 <sup>20</sup>
Bellman-Ford	Graph500 MPI	Lomonosov	18,0	SSCA-2	2 <sup>20</sup>
Dijkstra's	PBGL MPI	IBM BlueGene/P	8,9	SSCA-2	2 <sup>20</sup>
Delta-Stepping	PBGL MPI	IBM BlueGene/P	3,8	SSCA-2	2 <sup>20</sup>
Delta-Stepping	PBGL MPI	IBM BlueGene/P	1,3	RMAT	2 <sup>20</sup>
Dijkstra's	PBGL MPI	IBM BlueGene/P	0,6	RMAT	2 <sup>20</sup>

# Анализ приложений для решения **графовых задач** (В каких случаях данная компьютерная платформа эффективна?)



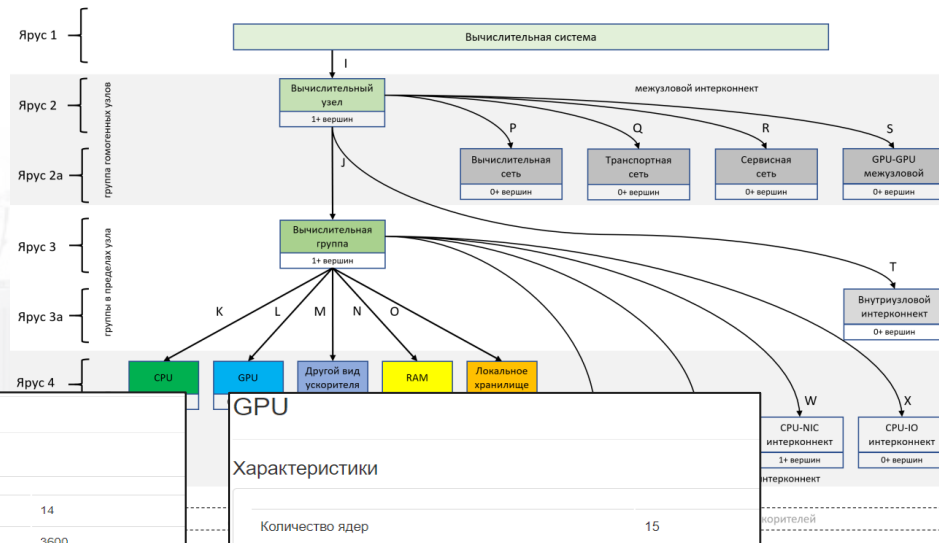
Problem	Algorithm	Implementation	Platform	MTEPS	GraphType	GraphSize
Breadth-first search	BFS	RCC for GPU	Lomonosov-2 (NVIDIA P100)	11061	RMAT	2 <sup>22</sup>
Breadth-first search	BFS	GAP	Lomonosov-2 (NVIDIA K40)	5350	RMAT	2 <sup>22</sup>
Breadth-first search	BFS	Ligra	Lomonosov-2 (NVIDIA K40)	4168	RMAT	2 <sup>22</sup>
Minimum Spanning Tree	Boruvka's	RCC for GPU	Lomonosov-2 (NVIDIA P100)	3793	RMAT	2 <sup>26</sup>
Single Source Shortest Path	Bellman-Ford	RCC for GPU	Lomonosov (NVIDIA 2090)	1309,0	SSCA-2	2 <sup>20</sup>
Strongly Connected Components	Shiloach-Vishkin	Ligra	Lomonosov-2 (x86)	1307,00	RMAT	2 <sup>26</sup>
Single Source Shortest Path	Bellman-Ford	Ligra	Lomonosov-2 (x86)	1035,0	RMAT	2 <sup>21</sup>
Single Source Shortest Path	Delta-Stepping	PBGL MPI	Cluster / "Angara" net	809,5	SSCA-2	2 <sup>21</sup>
Page Rank	Page Rank	Nvidia nvGraph	Lomonosov-2 (NVIDIA K40)	753	RMAT	2 <sup>18</sup>
Breadth-first search	BFS	RCC for CPU	NEC SX-ACE	715	RMAT	2 <sup>22</sup>
Strongly Connected Components	Forward-Backward	RCC for GPU	Lomonosov-2 (NVIDIA P100)	620,00	RMAT	2 <sup>21</sup>
Single Source Shortest Path	Delta-Stepping	GAP	Lomonosov-2 (x86)	616,0	RMAT	2 <sup>21</sup>
Strongly Connected Components	Forward-Backward	RCC for CPU	Lomonosov-2 (x86)	564,00	RMAT	2 <sup>24</sup>
Strongly Connected Components	Shiloach-Vishkin	GAP	Lomonosov-2 (x86)	547,00	RMAT	2 <sup>20</sup>
Minimum Spanning Tree	Boruvka's	RCC for GPU	Lomonosov (NVIDIA 2090)	204,441	SSCA-2	2 <sup>14</sup>
Page Rank	Page Rank	GAP	Lomonosov-2 (NVIDIA K40)	172	RMAT	2 <sup>24</sup>
Breadth-first search	BFS	PBGL MPI	IBM BlueGene/P	167	SSCA-2	2 <sup>22</sup>
Breadth-first search	BFS	PBGL MPI	Lomonosov (x86)	155	SSCA-2	2 <sup>22</sup>
Minimum Spanning Tree	Boruvka's	RCC for CPU	Intel Xeon Phi (KNL)	25,16	SSCA-2	2 <sup>21</sup>
Single Source Shortest Path	Dijkstra's	PBGL MPI	IBM BlueGene/P	8,9	SSCA-2	2 <sup>20</sup>

*Как связать свойства алгоритмов  
и особенности архитектуры  
вычислительных систем ?*

*Свойства алгоритмов описывать умеем – AlgoWiki.*

*Необходима модель описания структуры  
и параметров вычислительных систем ...*

# Модель описания вычислительных систем



CPU	
Характеристики	
Количество ядер	14
Максимальная тактовая частота, МГц	3600
Размер кэша L3, КБ	35840
Количество уровней кэш-памяти	3
Ширина векторных регистров, бит	512
Техпроцесс, нм	22
Средняя мощность, Ватт	145
Скорость системной шины, ГТ/с	9.6
Число каналов памяти	4
Пропускная способность памяти, ГБ/с	68
Пиковая производительность, Гфлоп/с	600
Тактовая частота (МГц)	2600
Размер кэша L1, КБ	32
Размер кэша L2, КБ	128
Модель CPU	Xeon E5-2697v3
Семейство CPU	Intel Xeon E5 (Haswell)
Поколение CPU	Intel Haswell
Производитель CPU	Intel
Технология многopotочности	Hyperthreading
Технология векторизации	AVX

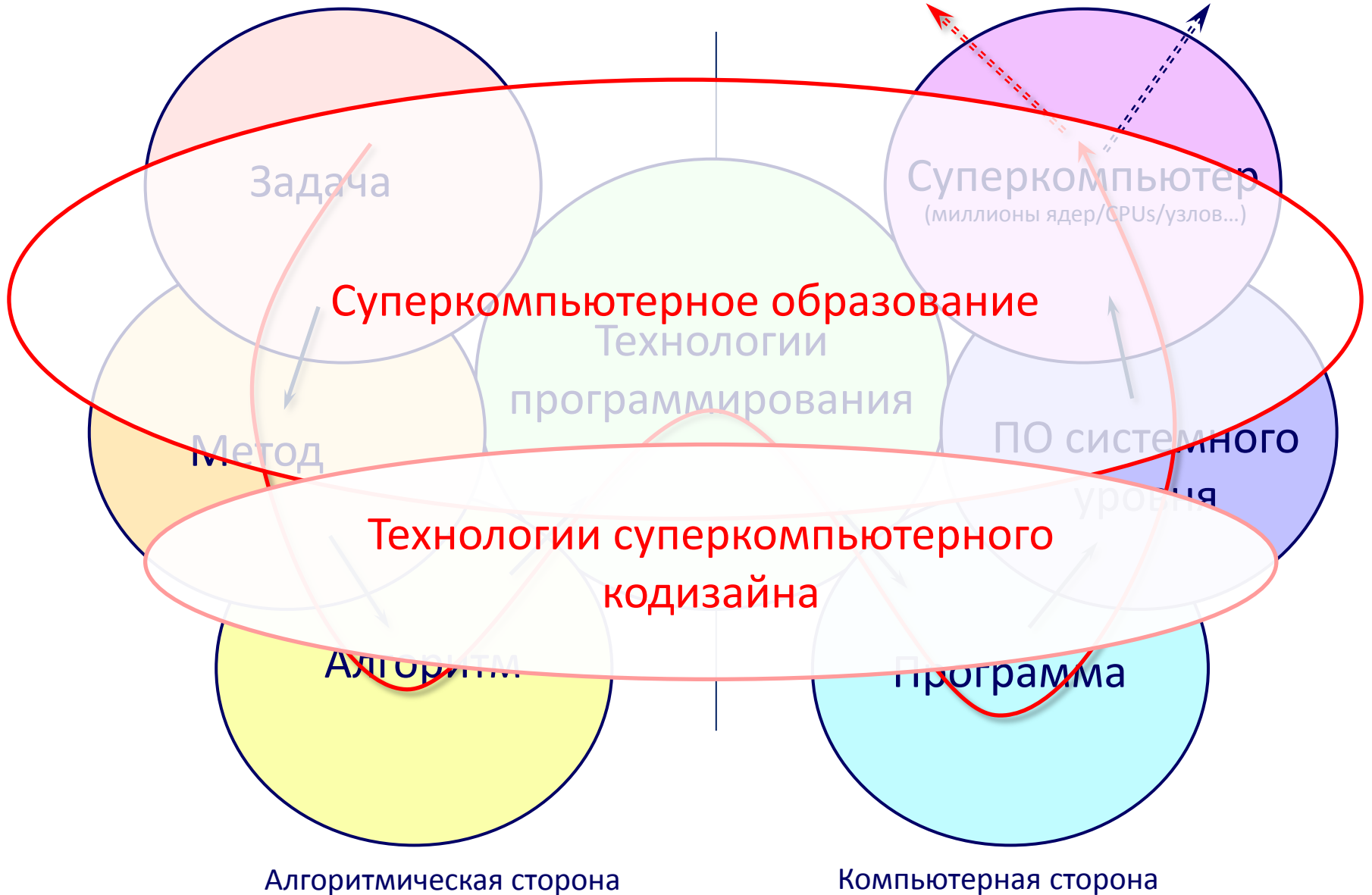
GPU	
Характеристики	
Количество ядер	15
Количество микроядер	2880
Тактовая частота (МГц)	667
Максимальная тактовая частота, МГц	745
Средняя мощность, Ватт	245
Техпроцесс, нм	28
Частота памяти, МГц	1502
Размер кэша L1, КБ	16
Размер кэша L2, КБ	1536
Количество уровней кэш-памяти	3
Объем памяти, Гб	12
Пропускная способность памяти, ГБ/с	288
Пиковая производительность, Гфлоп/с	1682
Модель GPU	Tesla K40M
Производитель GPU	NVIDIA
Архитектура GPU	Kepler
Тип внутренней памяти	GDDR5

Появляется возможность анализа, поиска и подбора по наборам конкретных параметров архитектуры вычислительных систем в связке со свойствами алгоритмической части.

Адекватная модель вычислительных систем – ключевая часть, открывающая возможность для конструктивного совместного анализа структуры алгоритмов и особенностей архитектуры.

# Решение задач и вычислительные системы

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





[Sarov.msu.ru](http://Sarov.msu.ru)





# Магистерские программы



## Направление «Физика»



**Теоретическая физика.** Руководители программы: Э. Э. Боос, чл.-корр. РАН, д.ф.-м.н., профессор  
А. В. Борисов, д.ф.-м.н., профессор



**Лазерная нелинейная оптика и фотоника.** Руководитель программы: В. А. Макаров, д.ф.-м.н., профессор



**Экстремальные электромагнитные поля, релятивистская плазма и аттосекундная физика.**  
Руководитель программы: А. Б. Савельев-Трофимов, д.ф.-м.н., профессор

## Направление «Математика»



**Суперкомпьютерные технологии и фундаментальная информатика.** Руководитель программы:  
Воеводин Владимир Валентинович, чл.-корр. РАН, д.ф.-м.н., профессор



**Вычислительные методы и методика моделирования.** Руководители программы: Е. Е. Тыртышников,  
академик РАН, Б. Н. Четверушкин, академик РАН



МГУ Саров

# Профессорско-преподавательский состав

5 академиков РАН,

8 член-корреспондентов РАН,

3 профессора РАН,

42 доктора наук,

68 кандидатов наук,

15 преподавателей, имеющих государственные премии РФ, премии Президента РФ, премии Правительства РФ, научные премии.

Ведущие преподаватели МГУ имени М.В.Ломоносова приезжают в МГУ Саров для проведения учебных занятий.



# Решение научных проблем мирового уровня



МГУ Саров



Создание открытого образовательного кластера фундаментальных исследований с мощнейшей экспериментальной и вычислительной базой.



Проведение научных исследований на уникальных установках.

## Основные проекты:

- синтез новых сверхтяжелых элементов;
- уточнение Стандартной модели;
- получение термоядерного зажигания;
- создание цифровых двойников и высокоинтеллектуальных систем;
- разработка отечественных высокопроизводительных систем и суперкомпьютеров.



# Участие студентов в научных школах и международных конференциях



МГУ Саров

20-я Международная конференция «Оптика лазеров – 2022»

Всероссийская школа по лазерной физике и лазерным технологиям



Международная научная конференция студентов, аспирантов и молодых ученых «Ломоносов-2022»

Летняя научная школа по физике высоких энергий и ускорительной технике

XXIII Харитоновские научные чтения "Экстремальные состояния вещества. Детонация. Ударные волны."

Всероссийская школа-семинар по математическому моделированию на суперкомпьютерах экса- и зеттафлопной производительности



XXX International Conference on neutrino Physics and Astrophysics

Научная школа НЦФМ по физике нейтрино

Международная научная конференция "Суперкомпьютерные дни в России"

II Конгресс молодых ученых в г. Сочи



# Студенты МГУ Саров на II Конгрессе молодых ученых в Сириусе



Участие студентов в интеллектуальной игре Что? Где? Когда?



Организация и участие в дискуссии: «СНО как механизм привлечения талантливых студентов в науку»



Участие в открытом диалоге “Быстрый старт НЦФМ” с участием Лихачева А.Е., Садовниченко В. А., Сергеева А. М.



## Стипендия студентов – 55 тыс.руб./мес.:

- обучение на хорошо и отлично,
- отсутствие пересдач,
- выполнение учебного плана по научной работе.

## Стипендия аспирантов – 75 тыс.руб./мес.

- встречи с выдающимися учеными и известными людьми;
- стажировки и ознакомительные поездки на предприятия Госкорпорации «Росатом», в институты РАН и др.;
- экскурсии по достопримечательностям Московской и Нижегородской областей и по другим интересным местам;
- летний отдых на базах отдыха МГУ и РФЯЦ-ВНИИЭФ.



# Прием в филиал и направления подготовки в 2023 году: магистратура, аспирантура



Направление подготовки

Бюджетные места



## Магистратура



Прикладная математика  
и информатика (01.04.02)

39



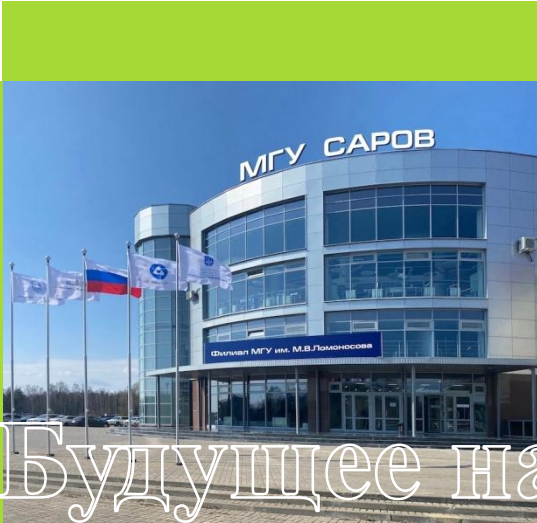
Физика (03.04.02)

40



## Аспирантура

– 20 целевых мест (бюджетные места)



МГУ Саров

Филиал Московского государственного университета  
имени М. В. Ломоносова в городе Сарове

Будущее начинается здесь! Будущее

Мы в социальных

СЕТЯХ: sarov\_msu\_ru



Оставьте заявку, чтобы  
не пропустить новости приемной  
кампании:

[exam.sarov.msu.ru](http://exam.sarov.msu.ru)





# Суперкомпьютерные дни в России 2023

международная научная конференция

25 - 26 сентября

два дня — множество параллельных событий

Семинар в рамках конференции:  
«Высшее образование  
для цифрового будущего:  
конвергенция HPC, Big Data, ML и IoT»

Научные школы

Суперкомпьютерная Академия

23.09 – 30.09.2023



Премия  
имени В.П.Гергея  
за лучшие молодежные доклады

*Благодарю за внимание !*

*[voevodin@parallel.ru](mailto:voevodin@parallel.ru)*