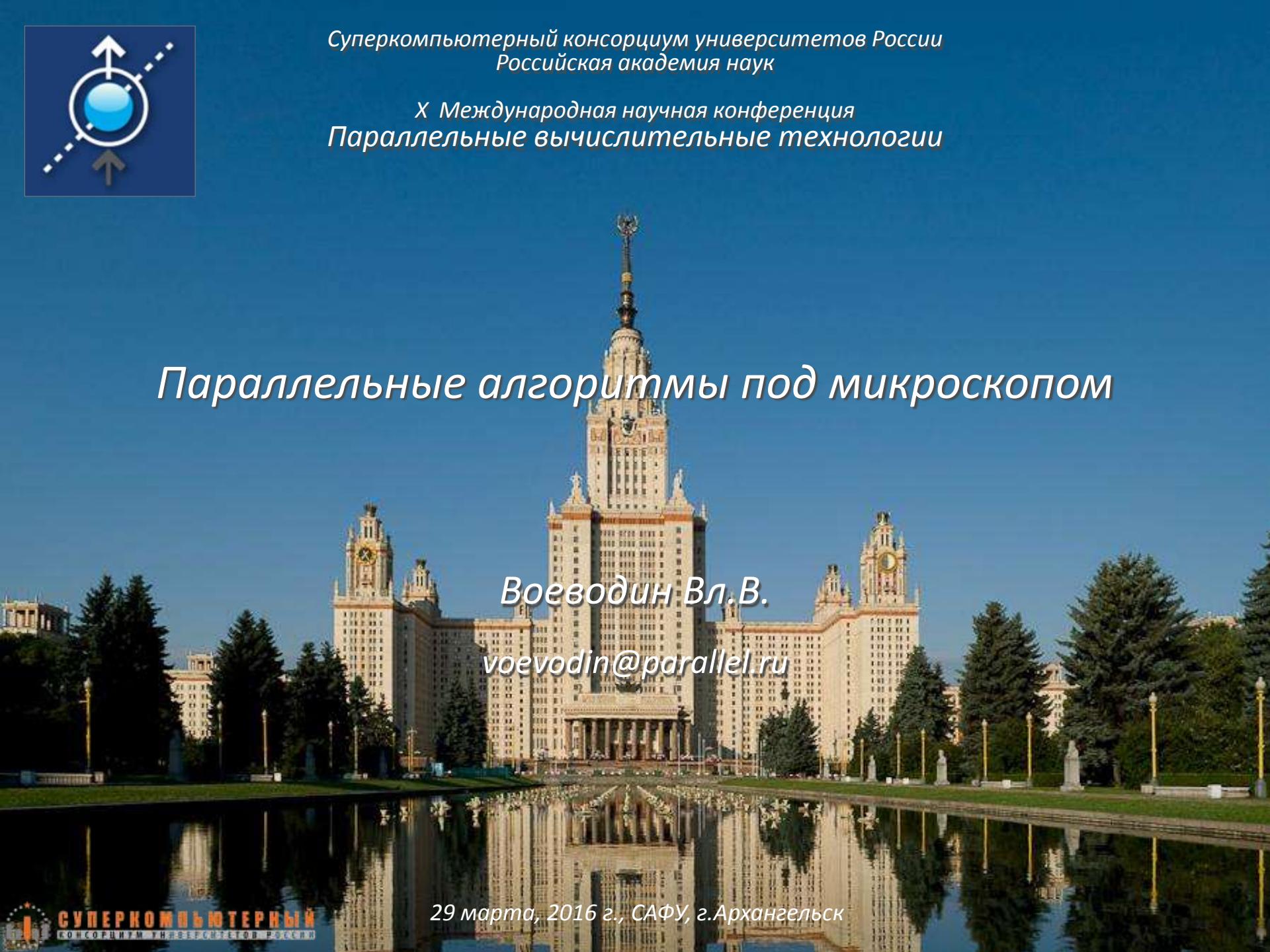


Суперкомпьютерный консорциум университетов России
Российская академия наук

X Международная научная конференция
Параллельные вычислительные технологии

Параллельные алгоритмы под микроскопом

Воеводин Вл.В.

voevodin@parallel.ru

29 марта, 2016 г., САФУ, г.Архангельск

СУПЕРКОМПЬЮТЕРНЫЙ
КОНСОРЦИУМ УНИВЕРСИТЕТОВ РОССИИ

Суперкомпьютер МГУ “Ломоносов-2”

Суперкомпьютерный центр МГУ сегодня:
Пользователи: 2955
Проекты: 880

Факультеты / Институты МГУ: 21
Институты РАН: 95
Университеты России: 102

1 стойка = 256 узлов: Intel Xeon (14c) + NVIDIA K40= 515 Tflop/s
Суперкомпьютер “Ломоносов-2” (5 стоек) = 2.5 Pflop/s

Суперкомпьютеры “Ломоносов” и “Ломоносов-2”: инфраструктура ПО последних лет

(основные элементы)

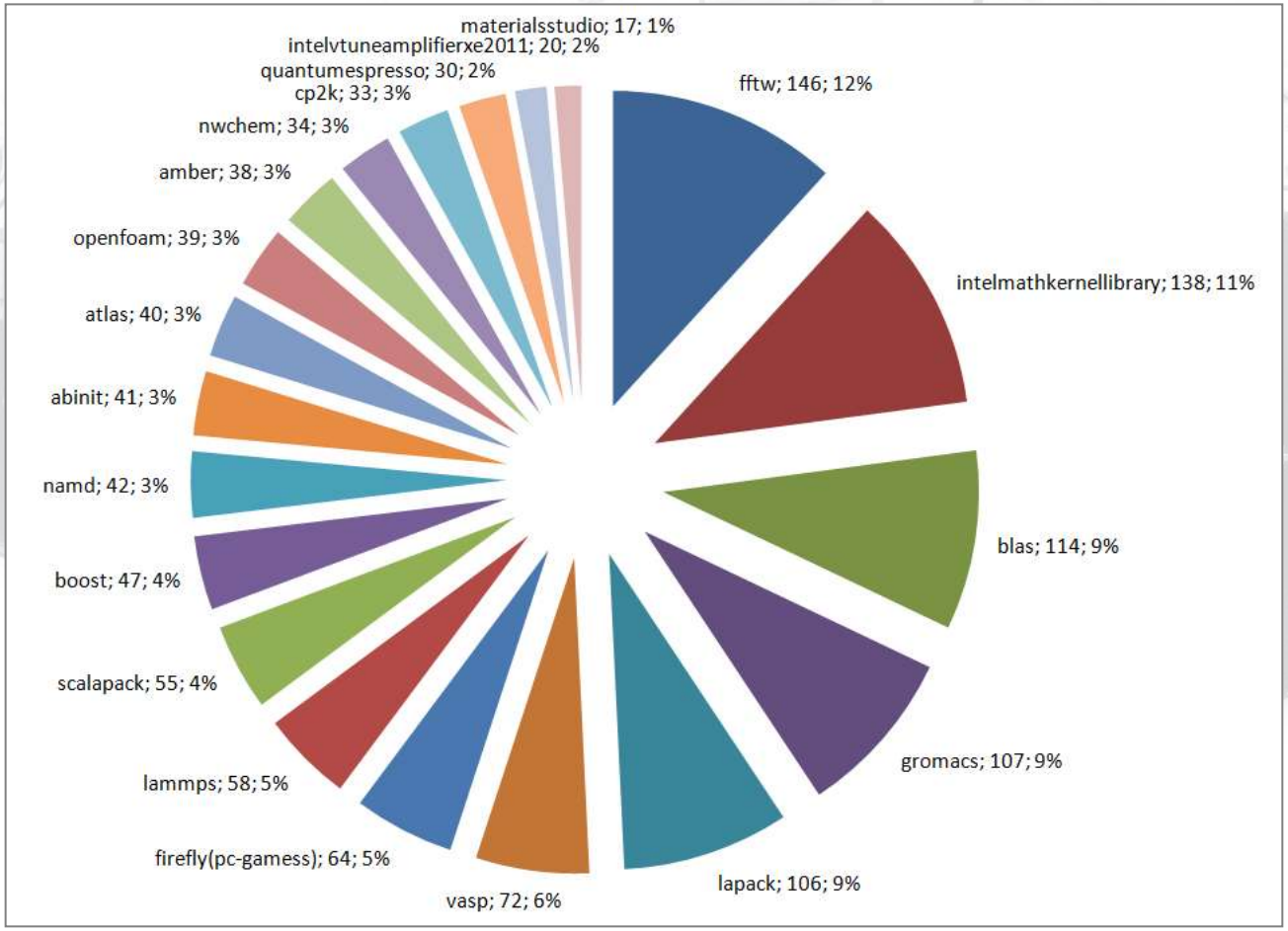
Прикладные пакеты, системы, инструменты разработки, библиотеки – 60+, и это число постоянно растет:

Intel ICC/IFORT, GCC, PathScale, PGI, MPIs, Intel VTune Performance Analyzer, Intel Cluster Tools, RogueWave TotalView, RogueWave ThreadSpotter, Allinea DDT, ScaLAPACK, ATLAS, IMKL, AMCL, BLAS, LAPACK, FFTW, cuBLAS, cuFFT, MAGMA, cuSPARSE, CUSP, and cuRAND…

VASP, WIEN2k, CRYSTAL, Gaussian, MOLPRO, Turbomole, Accelrys Material Studio, MesoProp, MOLCAS, Gromacs, FireFly, LAMMPS, NAMD, GAMESS, Quantum ESPRESSO, ABINIT, Autodock, CP2K, NWChem, PRIRODA, SIESTA, Amber, CPMD, DL POLY, VMD, GULP, Aztec, Geant, OpenFOAM, PARMETIS, FDMNES, GSL, METIS, Msieve, Octave, OpenMX, PETSc, SMEAGOL, VisIt, VTK, WRF…

Суперкомпьютеры “Ломоносов” и “Ломоносов-2”: инфраструктура ПО последних лет

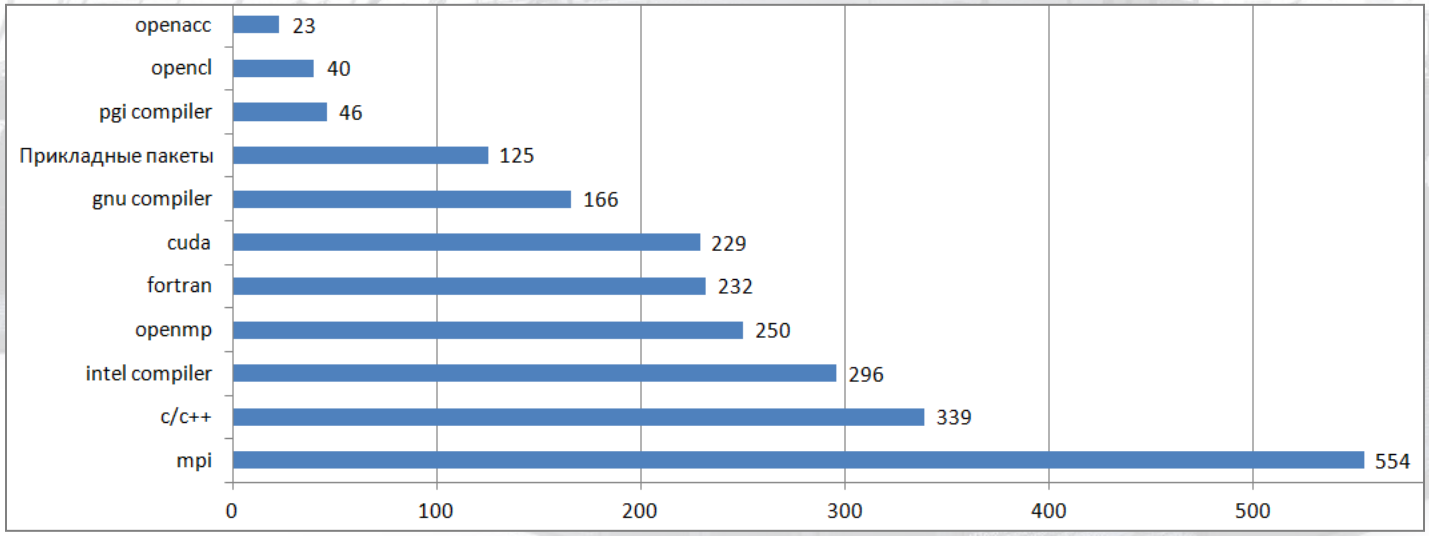
(краткая статистика)



Инфраструктура ПО обширна. А какова масштабируемость приложений?

Суперкомпьютеры “Ломоносов” и “Ломоносов-2”: инфраструктура ПО последних лет

(краткая статистика)



Большое число средств разработки ПО.

А какова масштабируемость приложений?

Тонкий анализ суперкомпьютерных приложений: динамика исполнения

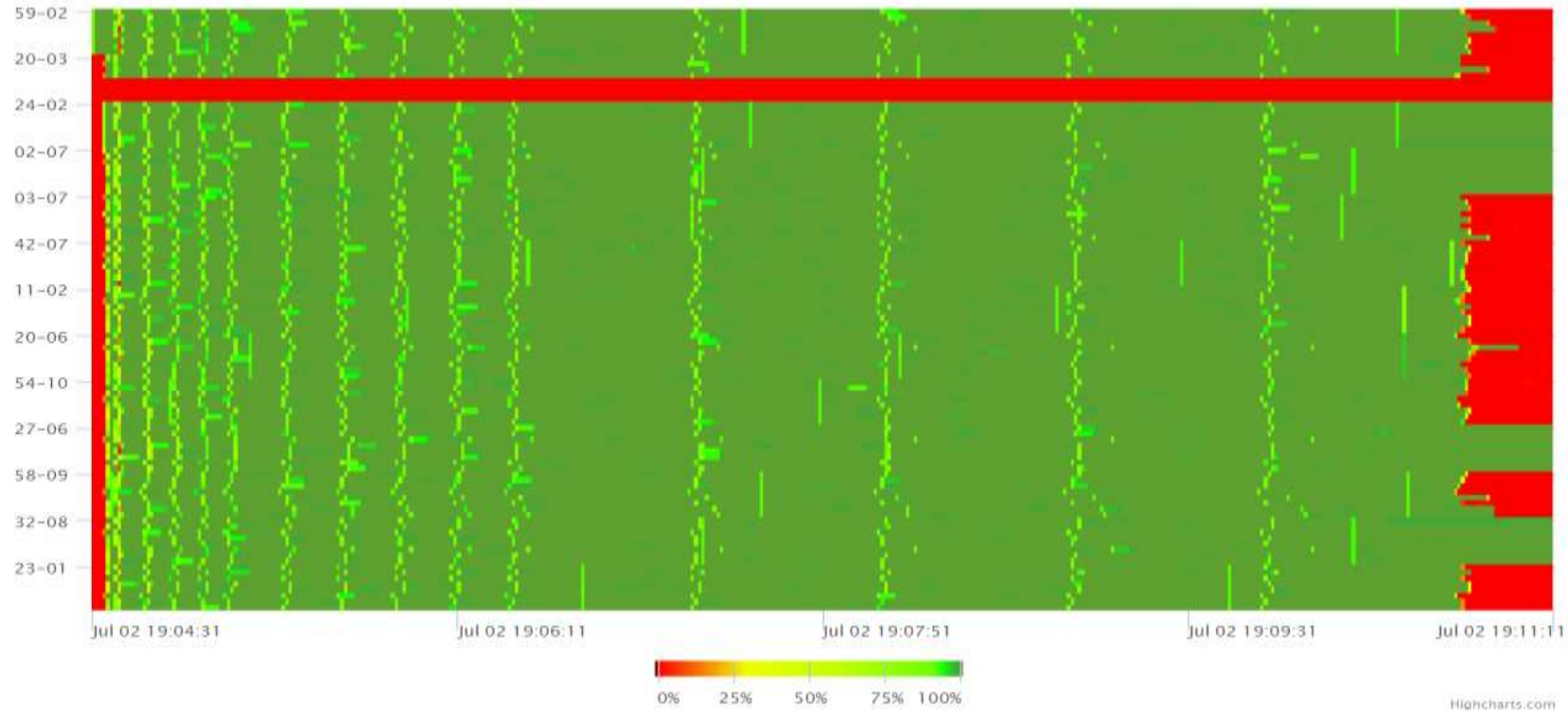
Информация о задаче № regular-1404302831-496016 пользователя

expand all

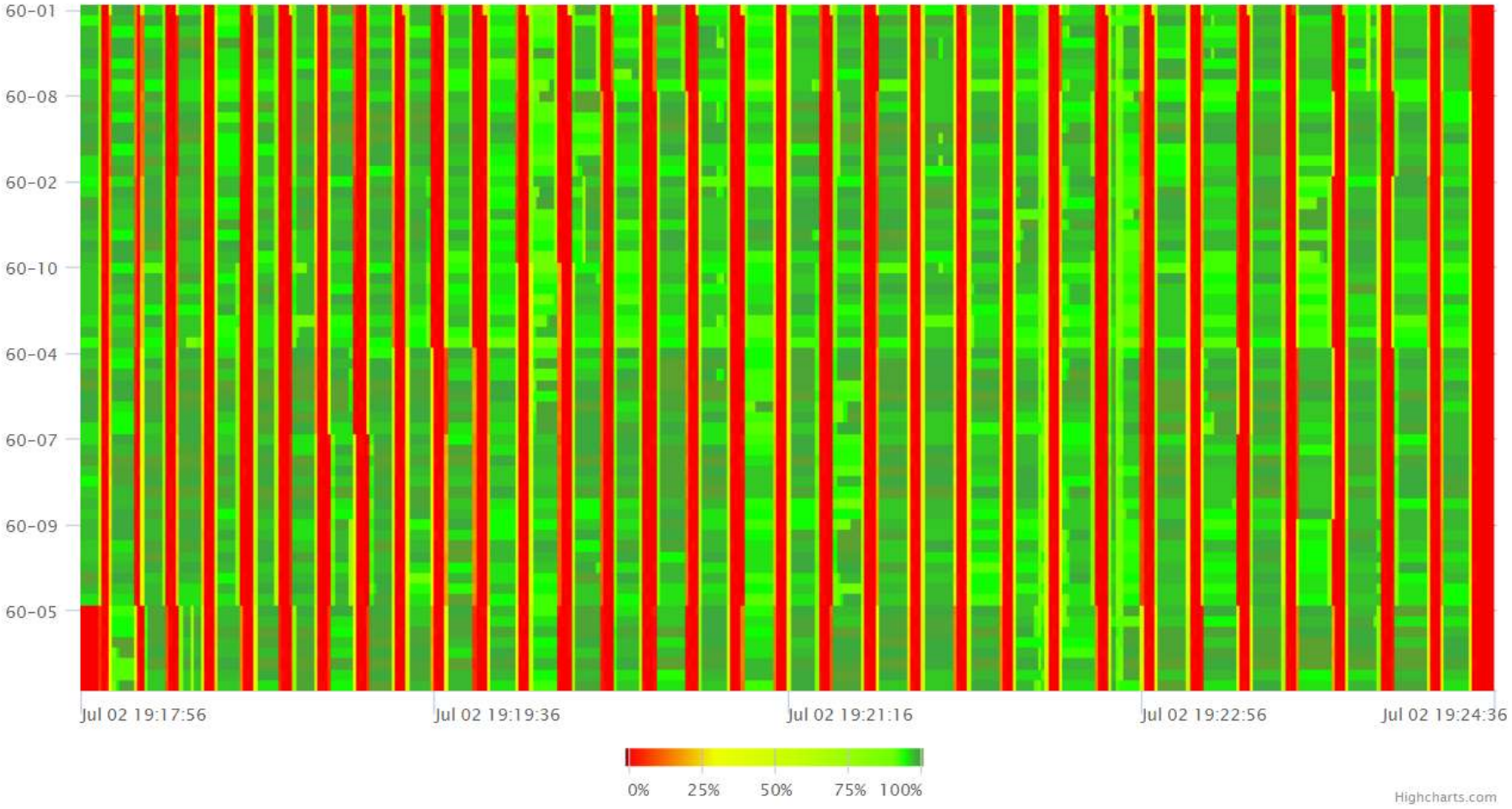
Информация о запуске

Строка запуска:	./2dxy_m01_p08.ex
Число ядер:	100
Номера узлов:	node-02-07,node-03-07,node-11-02,node-20-03,node-20-06,node-23-01,node-24-02,node-27-06,node-32-08,node-42-07,node-54-10,node-58-09,node-59-02
Дата постановки в очередь:	Wed, 02 Jul 2014 16:07:11 (1404302831)
Дата запуска:	Wed, 02 Jul 2014 19:04:29 (1404313469)
Дата окончания счета:	Wed, 02 Jul 2014 19:46:13 (1404315973)
Время счета:	0 days 0 hours 41 minutes 44 seconds
Время ожидания:	0 days 2 hours 57 minutes 18 seconds

Количе
процес



Тонкий анализ суперкомпьютерных приложений: динамика исполнения



Тонкий анализ суперкомпьютерных приложений: динамика исполнения

Хорошо ли пользователи знают свойства, особенности, статические и динамические характеристики параллельных алгоритмов?

А должны ли пользователи думать о параллельных алгоритмах?

Да…

Степень параллелизма

2005	2016	2025	
10^4	10^6	10^9	Суперкомпьютеры
2-4	12-64	10^4	Серверы…
1	4-8	10^3	ПК, ноутбуки…
1	1-4	10^2	Планшеты, смартфоны…

Хорошо ли пользователи знают свойства, особенности, статические и динамические характеристики параллельных алгоритмов?

А должны ли пользователи думать о параллельных алгоритмах?

Да…

Насколько хорошо пользователи знают архитектуру параллельных компьютеров ?

А должны ли пользователи думать об архитектуре?

Да… К сожалению, Да…

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)
…

Виден ли конец процессу переписывания программ?..

Можно ли выполнить такой анализ “раз и навсегда” ?

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…

Описание алгоритмов
(Что должно быть учтено в подобном описании?)

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

Описание алгоритмов
(Что должно быть учтено в подобном описании?)

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

Алгоритмы:
теоретический потенциал
(машинно-независимые свойства)

Алгоритмы:
особенности реализации

Описание алгоритмов
(на примере разложения Холецкого)

Кратко о важном

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: $\frac{n(n+1)}{2}$

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 исключительно для плотных симметричных положительно определенных матриц. В настоящее время его использование гораздо шире. Оно может быть применено также, например, к эрмитовым матрицам. Для повышения производительности вычислений часто применяется блочная версия разложения.

Для разреженных матриц разложение Холецкого также широко применяется в качестве основного этапа прямого метода решения линейных систем. В этом случае используют специальные упорядочивания для уменьшения ширины профиля исключения, а следовательно и уменьшения количества арифметических операций. Другие упорядочивания используются для выделения независимых блоков вычислений при работе на системах с параллельной организацией.

1.2 Математическое описание алгоритма

Исходные данные: положительно определённая симметрическая матрица A (элементы a_{ij}).

Вычисляемые данные: нижняя треугольная матрица L (элементы l_{ij}).

Формулы метода:

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

Описание алгоритмов
(на примере разложения Холецкого)

Общее описание

For positive definite Hermitian matrices (*symmetric matrices in the real case*), we use the decomposition $A = LL^*$, where L is the lower triangular matrix, or the decomposition $A = U^*U$, where U is the upper triangular matrix. These forms of the Cholesky decomposition are equivalent in the sense of the amount of arithmetic operations and are different in the sense of data represntation. The essence of this decomposition consists in the implementation of formulas obtained uniquely for the elements of the matrix L from the above equality. The Cholesky decomposition is widely used due to the following features.

Математическое описание

Input data: a symmetric positive definite matrix A whose elements are denoted by a_{ij}).

Output data: the lower triangular matrix L whose elements are denoted by l_{ij}).

The Cholesky algorithm can be represented in the form

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

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

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

$$l_{ji} = \left(a_{ji} - \sum_{p=1}^{i-1} l_{ip} l_{jp}\right) / l_{ii}, \quad i \in [2, n-1], j \in [i+1, n].$$

Комментарии к алгоритму

The Cholesky decomposition allows one to use the so-called *accumulation mode* due to the fact that the significant part of computation involves *dot product operations*. Hence, these dot products can be accumulated in double precision for additional accuracy. In this mode, the Cholesky method has the least *equivalent perturbation*. During the process of decomposition, no growth of the matrix elements can occur, since the matrix is symmetric and positive definite. Thus, the Cholesky algorithm is unconditionally stable.

Описание алгоритмов
(на примере разложения Холецкого)

Вычислительное ядро

A computational kernel of its serial version can be composed of $\frac{n(n-1)}{2}$ dot products of the matrix rows:

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

Последовательная сложность

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

- n square roots,
- $\frac{n(n-1)}{2}$ divisiona,
- $\frac{n^3-n}{6}$ multiplications and $\frac{n^3-n}{6}$ additions (subtractions): the main amount of computational work.

Базовая схема реализации

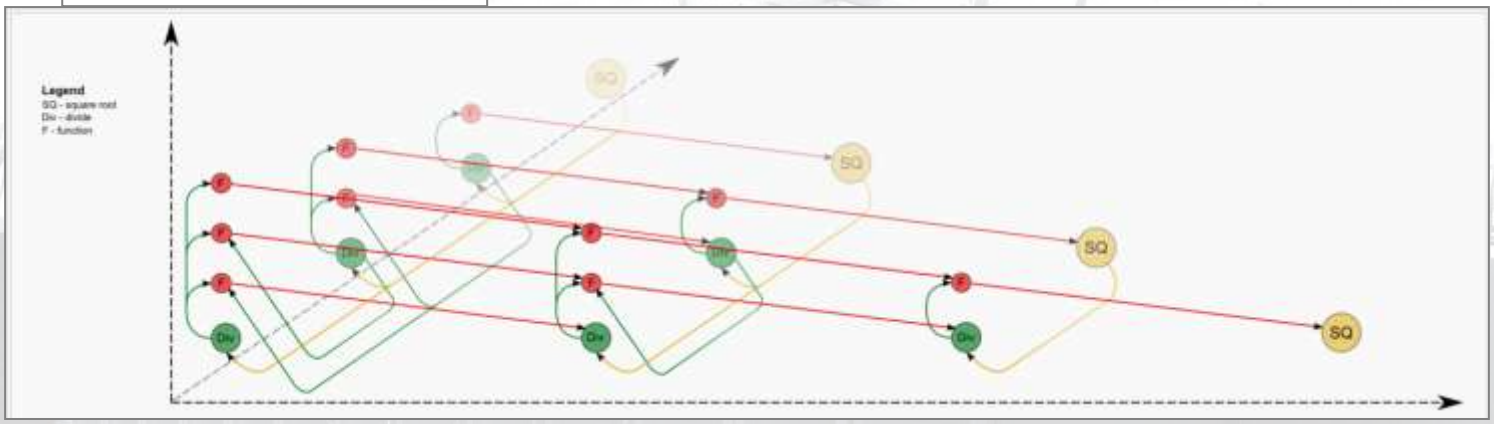
```
DO  I = 1, N
        S = A(I,I)
        DO  IP=1, I-1
                S = S - DPROD(A(I,IP), A(I,IP))
        END DO
        A(I,I) = SQRT (S)
        DO  J = I+1, N
                S = A(J,I)
                DO  IP=1, I-1
                        S = S - DPROD(A(I,IP), A(J,IP))
                END DO
                A(J,I) = S/A(I,I)
        END DO
END DO
```

Дополнительная информация

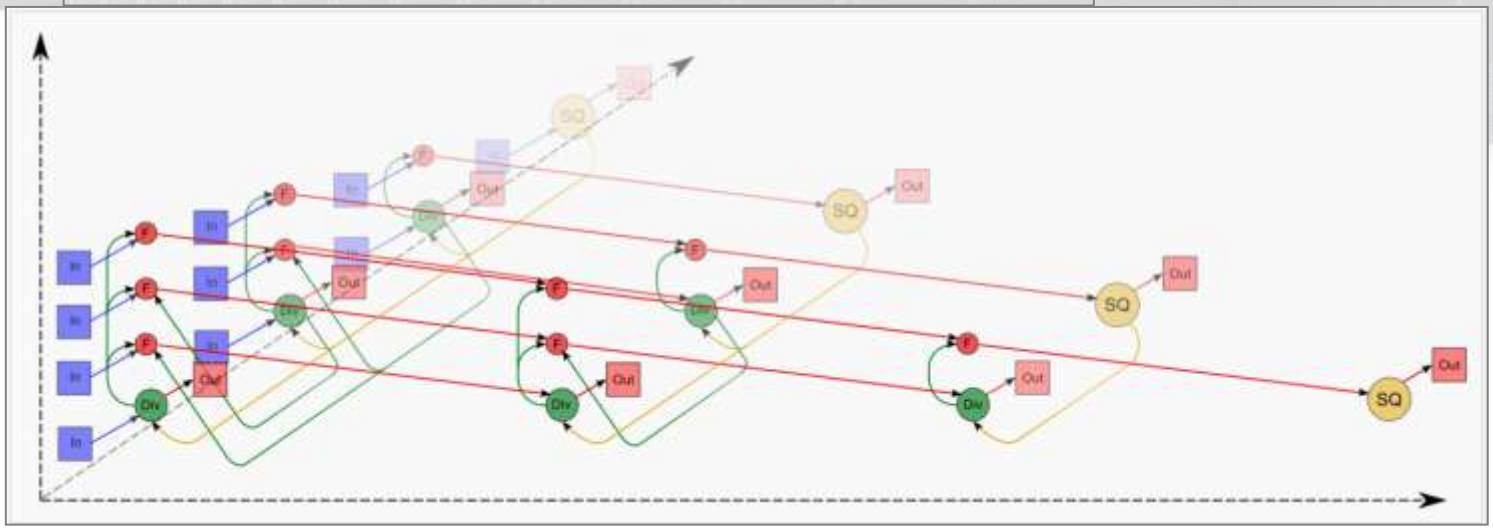
There exist block versions of this algorithm;

Описание алгоритмов
(на примере разложения Холецкого)

Информационная структура



Информационная структура с указанием входных и выходных данных



Описание алгоритмов
(на примере разложения Холецкого)

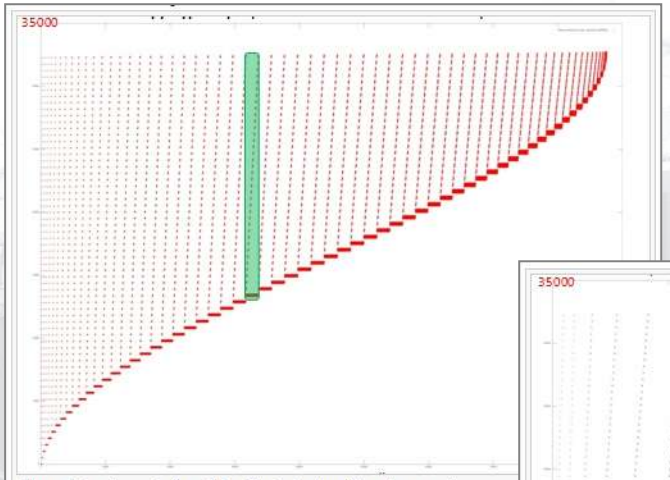


Figure 3. Implementation of the Cholesky algorithm. A general memory-

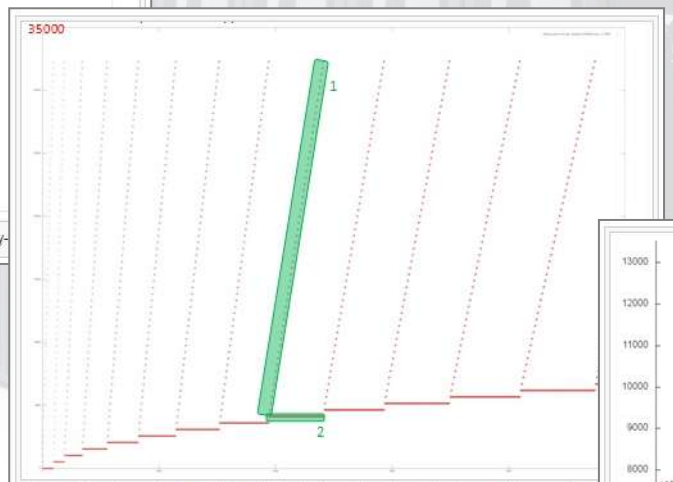


Figure 4. Implementation of the Cholesky algorithm. A profile fragment (several firs iterations).

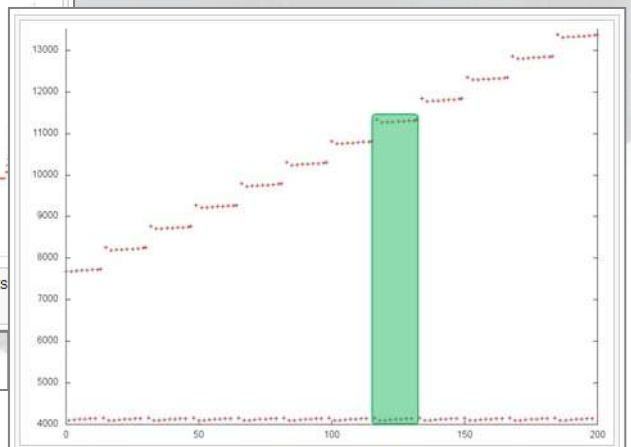
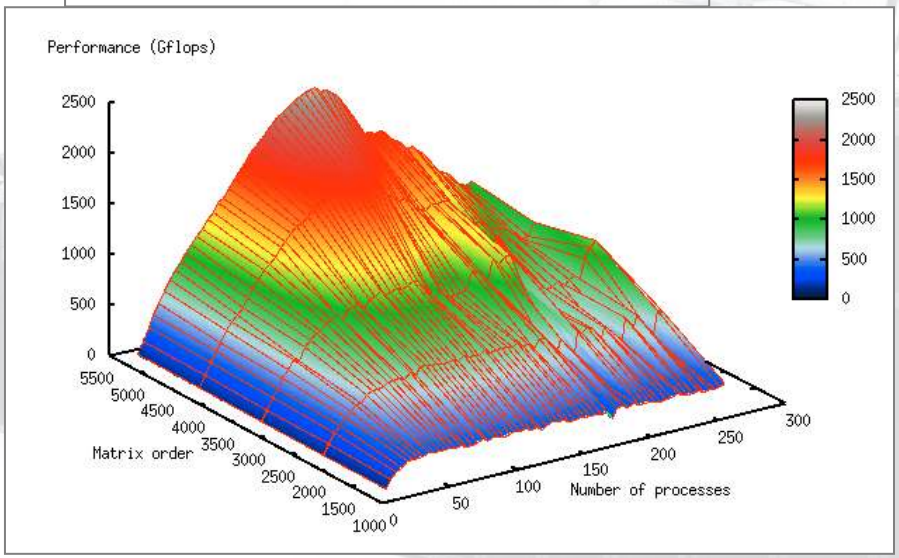


Figure 5. Implementation of the Cholesky algorithm. A profile fragment (a part of a single iteration).

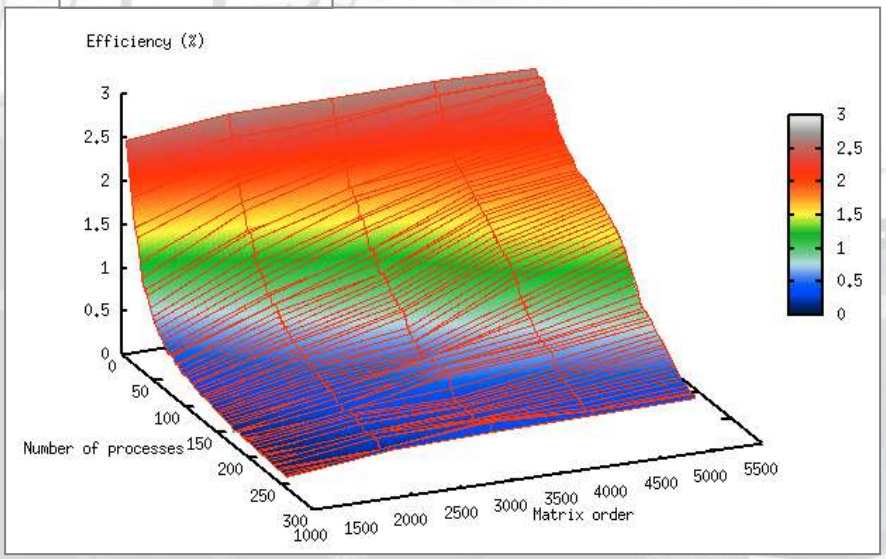
Локальность данных (профиль работы с памятью)

Описание алгоритмов
(на примере разложения Холецкого)

Масштабируемость (производительность) *



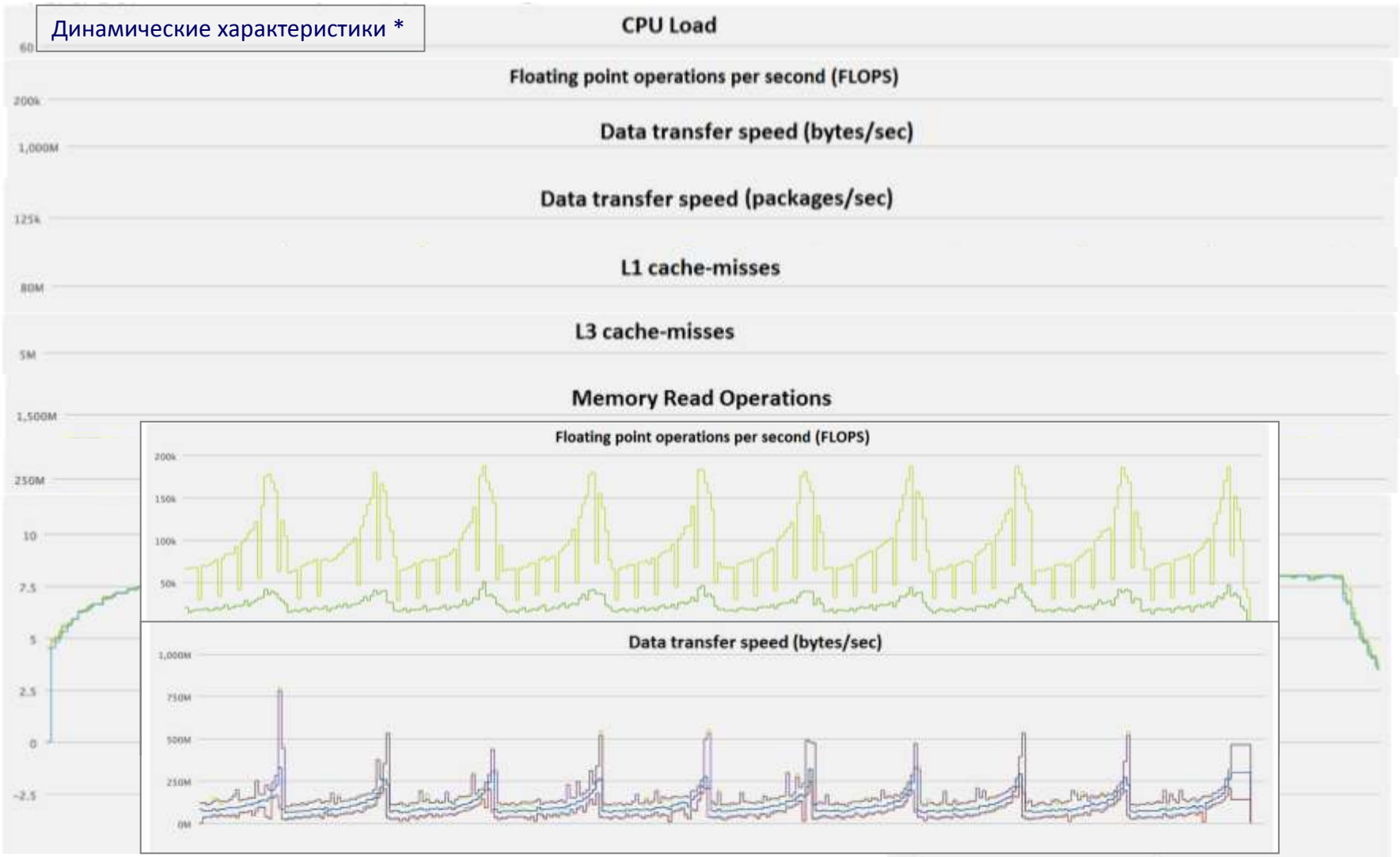
Эффективность *



* Масштабируемость, производительность, эффективность измерялись на суперкомпьютере МГУ “Ломоносов”.

Описание алгоритмов
(на примере разложения Холецкого)

Динамические характеристики *



* Динамические характеристики получены на суперкомпьютере МГУ “Ломоносов”.

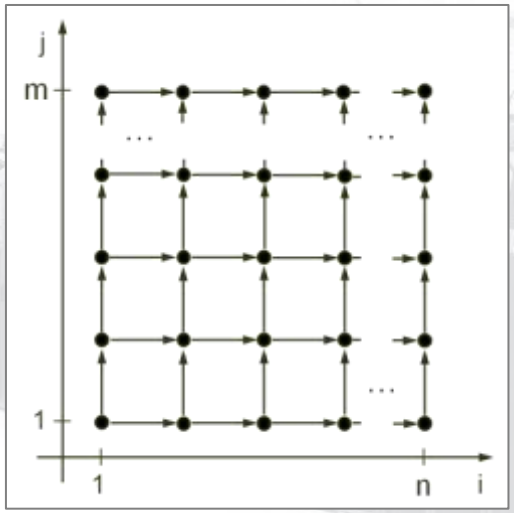
Это очень полезная информация об алгоритме, она действительно нужна.

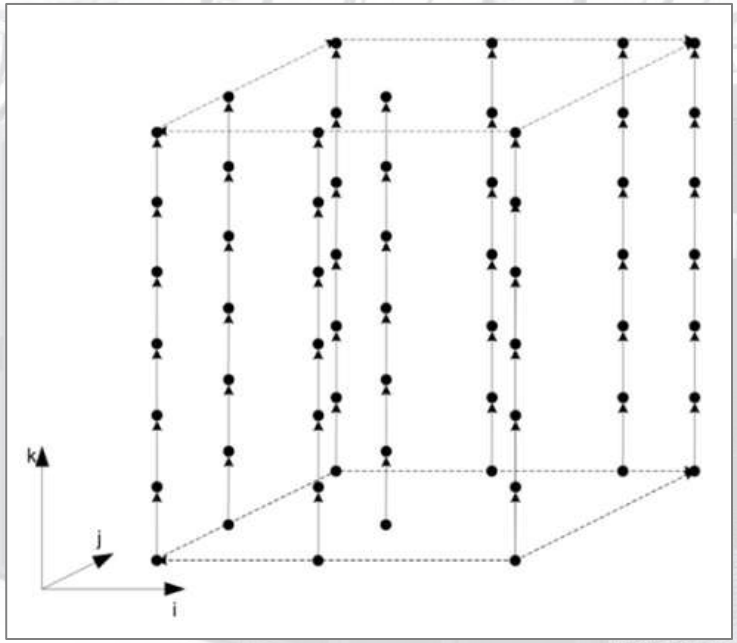
Но…

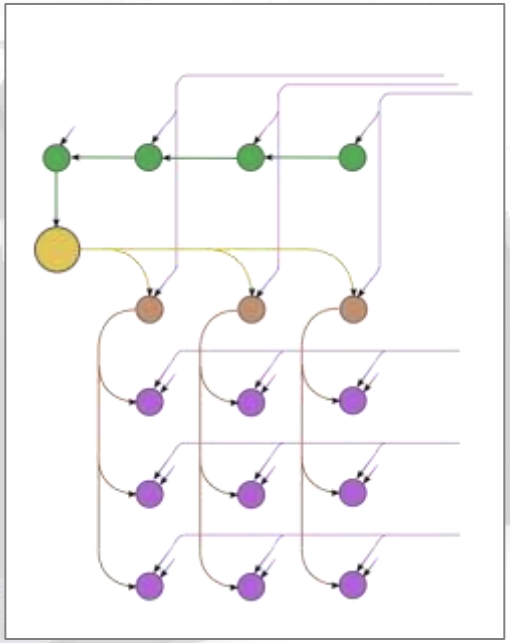
Создание полного описания алгоритма это не сложно. Это очень сложно…

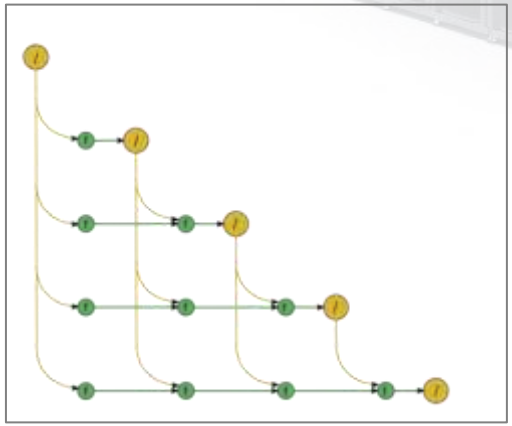
Информационная структура: как получать, описывать, показывать… ?

(сложности описания алгоритмов)





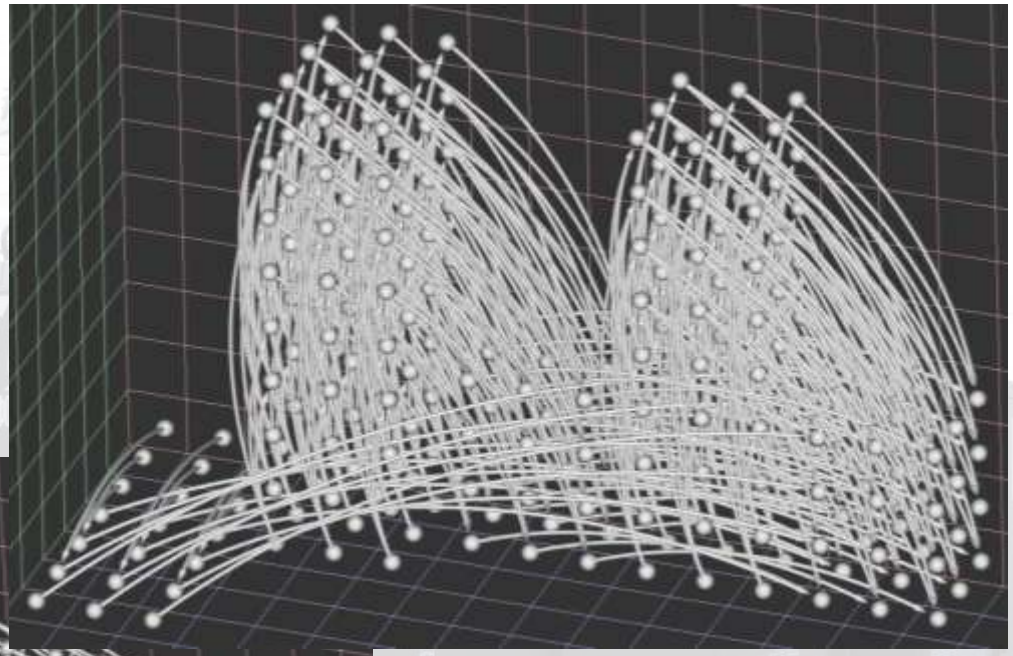


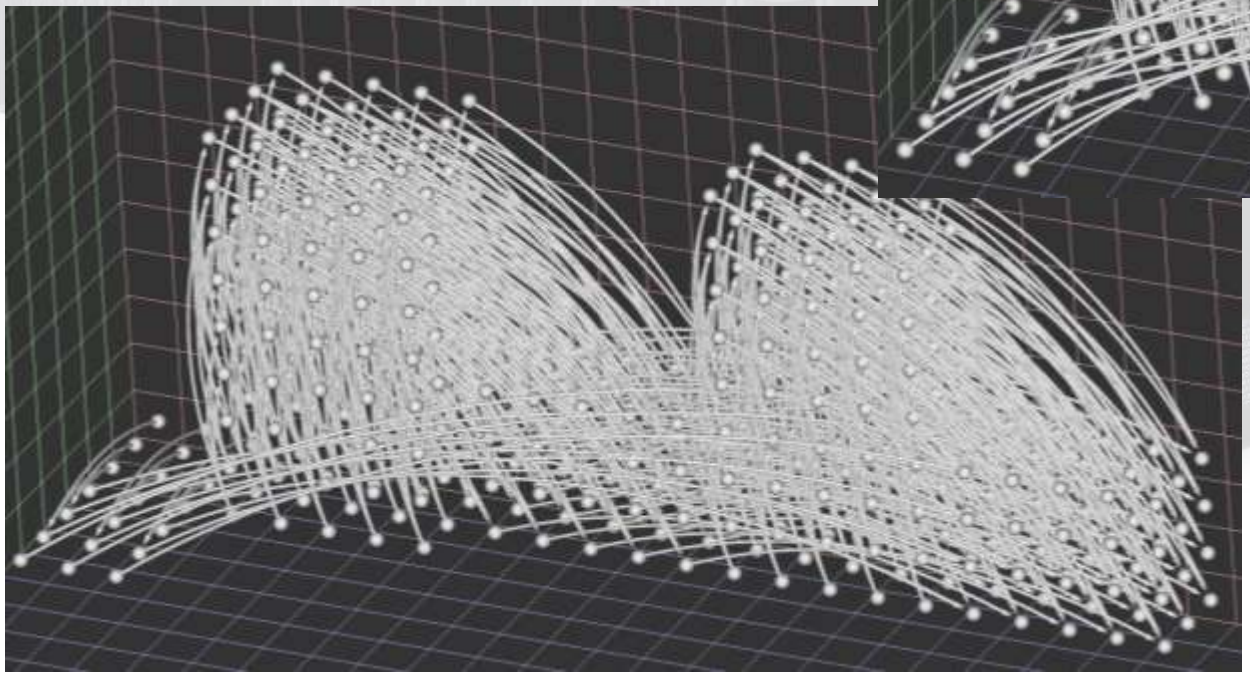


- Как изобразить потенциально бесконечный граф ?
- Как изобразить потенциально многомерный граф ?
- Как показать зависимость структуры графа от размера задачи ?

Информационная структура: как получать, описывать, показывать… ?

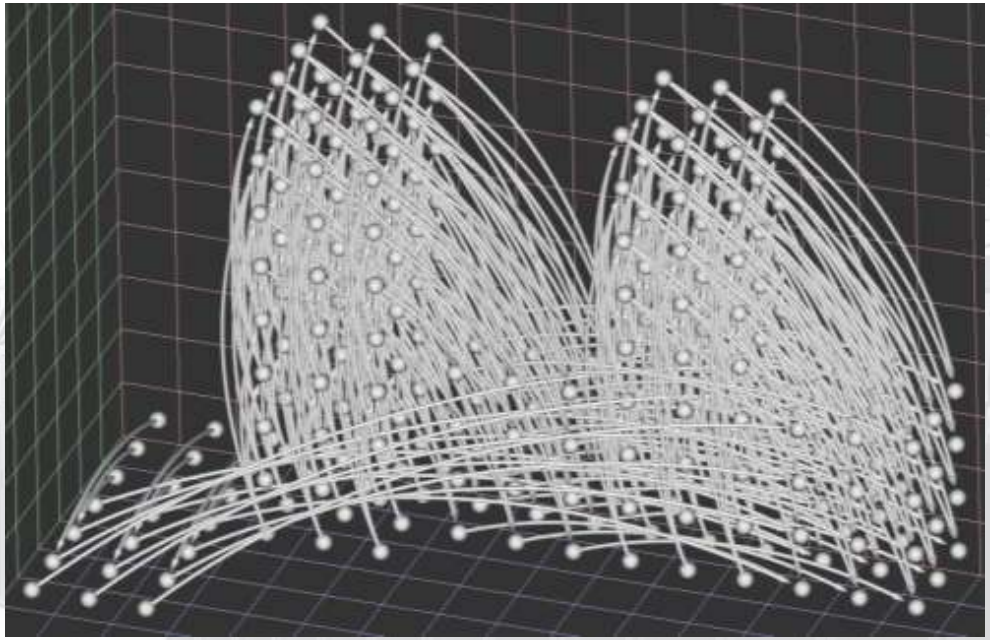
(сложности описания алгоритмов)

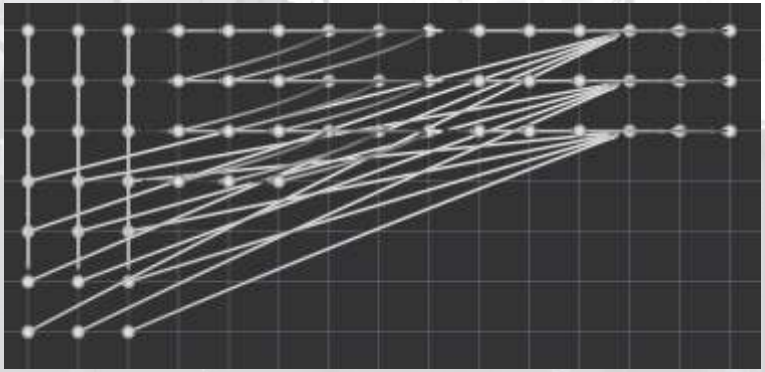


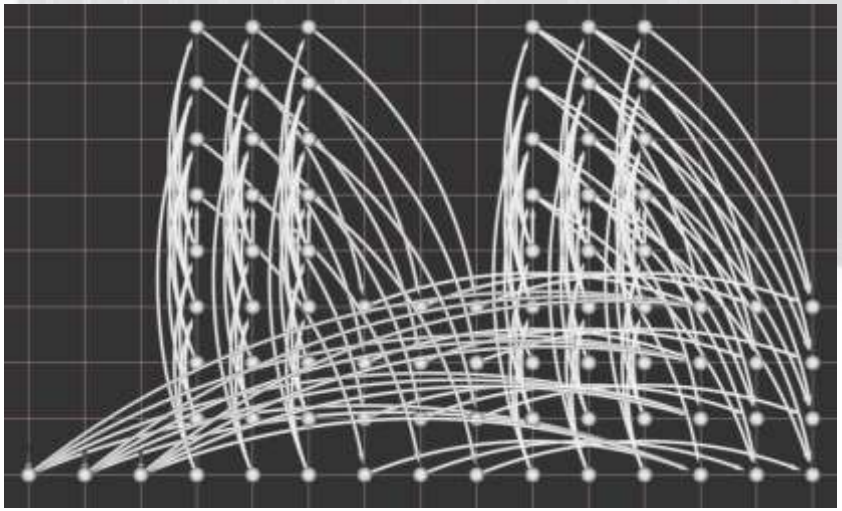


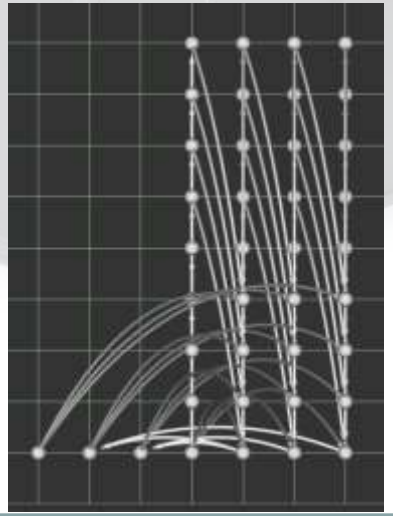
Информационная структура: как получать, описывать, показывать… ?

(сложности описания алгоритмов)









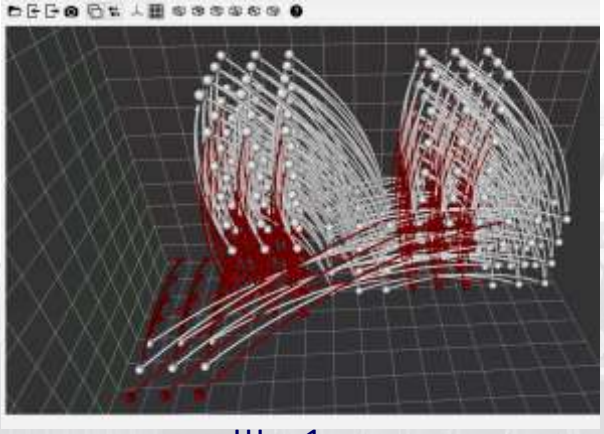
Информационная структура: как получать, описывать, показывать… ?

(сложности описания алгоритмов)

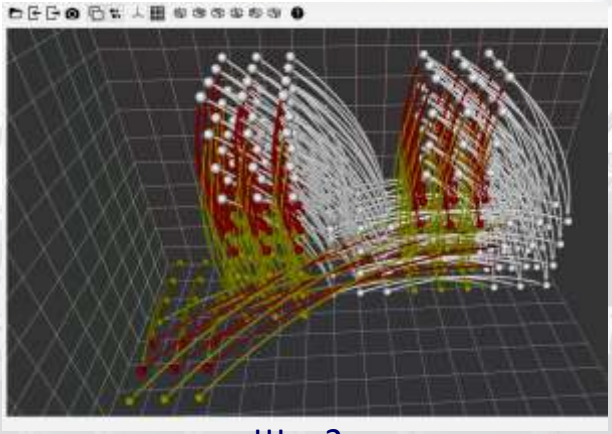
Как выразить имеющийся параллелизм и показать возможный способ параллельного исполнения ?

Информационная структура: как получать, описывать, показывать… ?

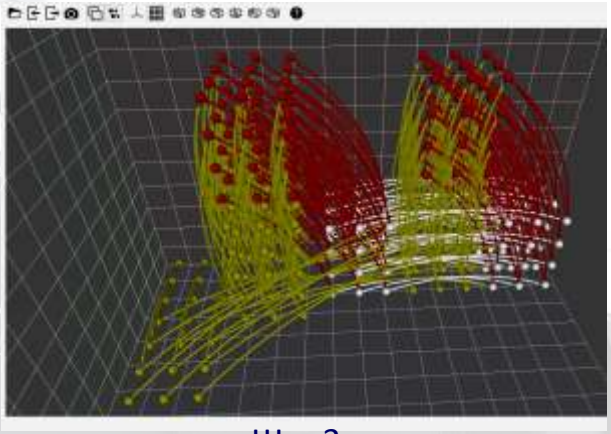
(сложности описания алгоритмов)



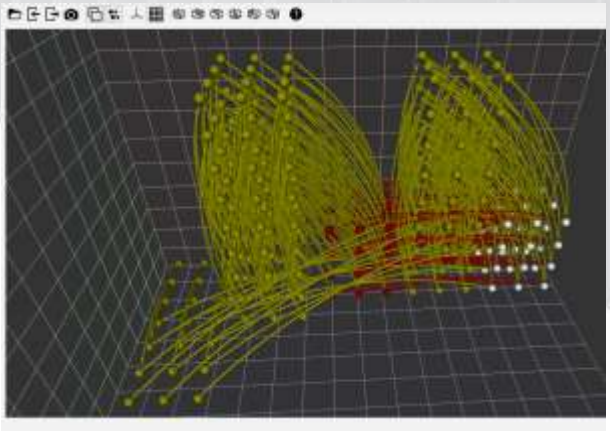
Шаг 1



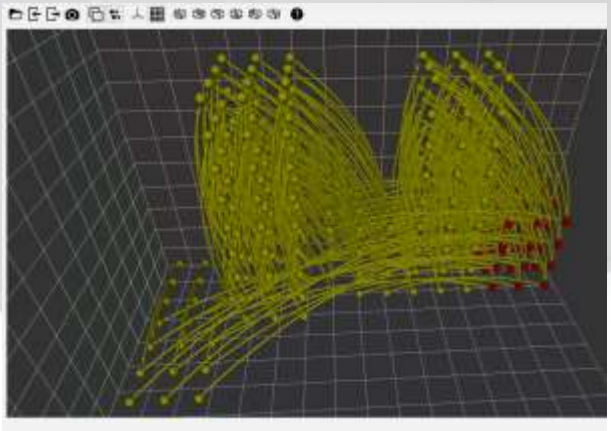
Шаг 2



Шаг 3



Шаг 4



Шаг 5

Цвета в ярусно-параллельной форме:

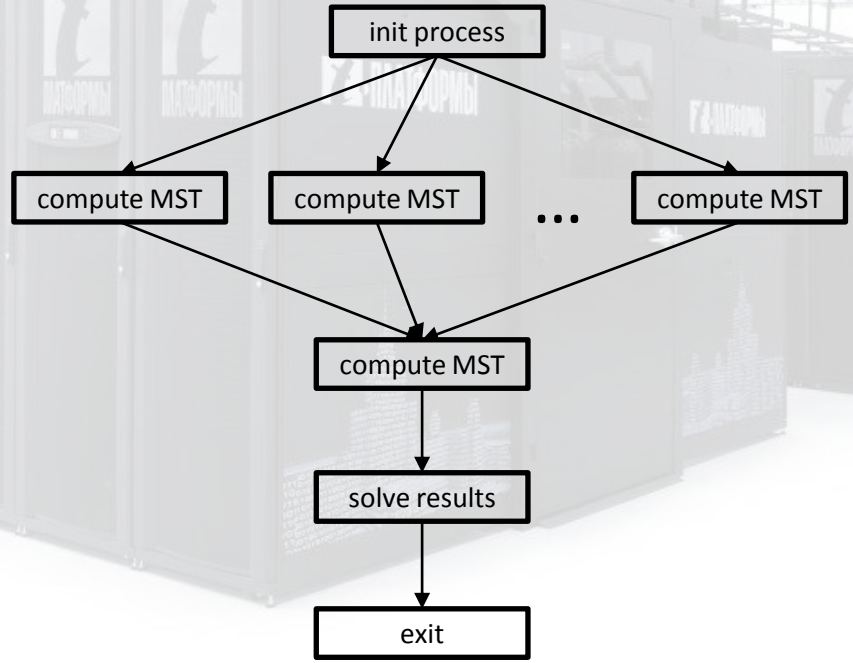
Красный – слой выполняющихся операций; **Зеленый** – уже выполненные операции;

Белый – операции должны выполниться позже.

Потенциальный параллелизм: как находить, описывать, показывать… ?

(сложности описания алгоритмов)

$$MST(E_1 \cup E_2) = MST\big(MST(E_1) \cup MST(E_2)\big),$$
$$E = E_1 \cup E_2$$

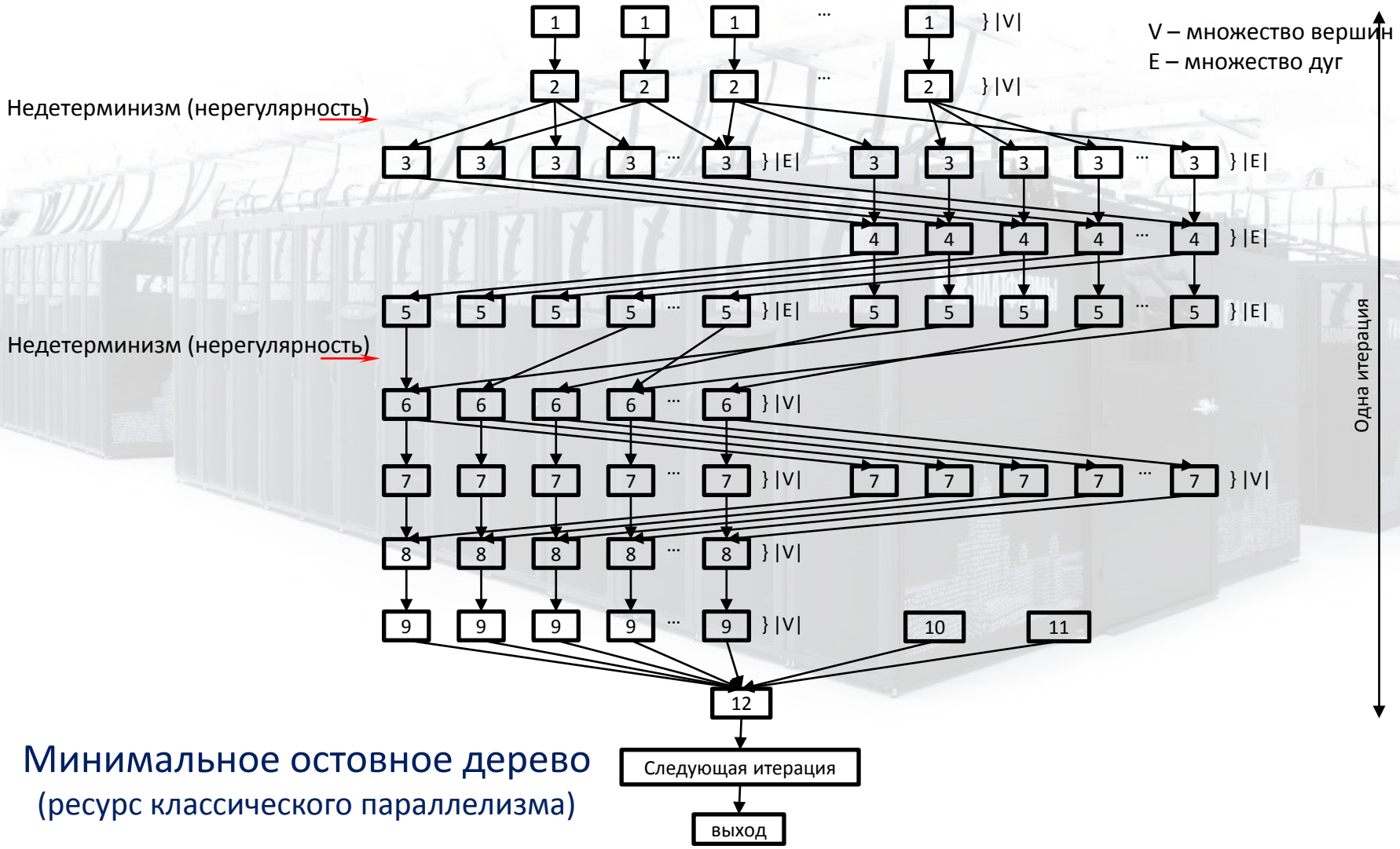


Минимальное остовное дерево

(ресурс “математического” параллелизма)

Потенциальный параллелизм: как находить, описывать, показывать… ?

(сложности описания алгоритмов)



Минимальное остовное дерево

(ресурс классического параллелизма)

Структура алгоритмов: возможные варианты для параллельного кода

(сложности описания алгоритмов)

Предположим, что мы знаем потенциальный параллелизм алгоритма...
Как выразить “потенциальную свободу” в выборе походящей формы для параллельной программы ?

Структура алгоритмов: возможные варианты для параллельного кода

(сложности описания алгоритмов)

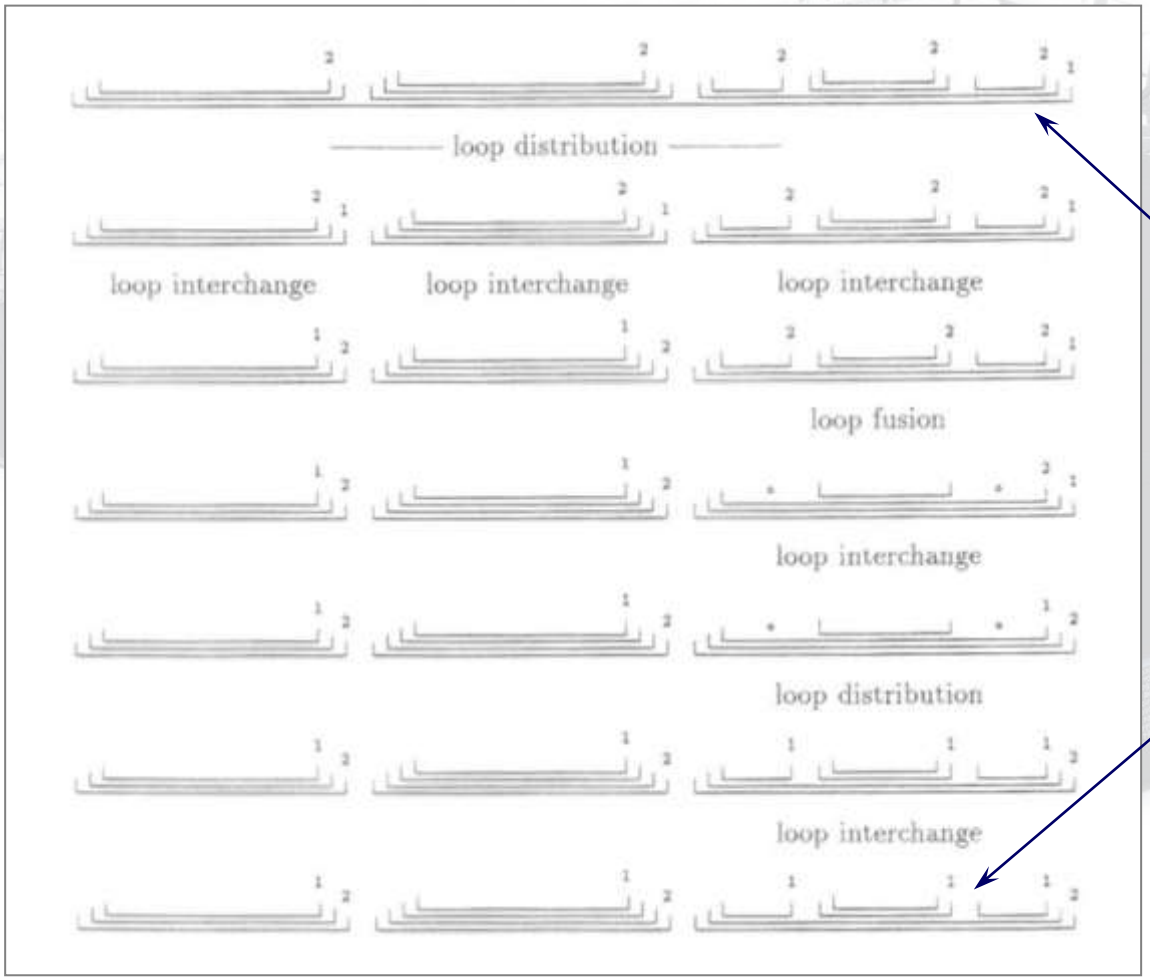
Циклический профиль алгоритма / программы

Самый внешний цикл (отмечен “1”) - параллельный.

Можно ли сделать данный цикл самым внутренним, чтобы векторизовать?

Структура алгоритмов: возможные варианты для параллельного кода

(сложности описания алгоритмов)



Эта последовательность преобразований делает исходный внешний цикл (1) внутренним и позволяет векторизовать программу (алгоритм).

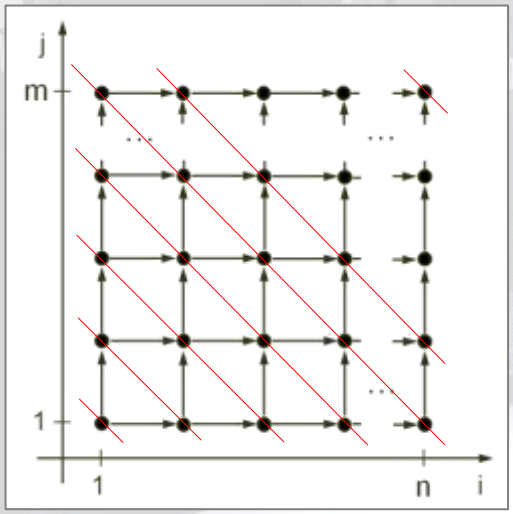
Как показать подобный потенциал преобразований в общем случае?

Возможные источники несбалансированности: это важно, это нужно отметить
(сложности описания алгоритмов)

Баланс между арифметическими операциями +/- and * ;

Баланс между арифметическими операциями и “чтениями/записями”;

Баланс в числе операций, которые могут выполняться параллельно;



Баланс между вычислительной и коммуникационной частями…

Возможные источники недетерминизма: это важно, это нужно отметить
(сложности описания алгоритмов)

Структура входных данных (разреженные матрицы, графы произвольной структуры…);

Итерационная природа алгоритмов;

Датчики случайных чисел;

Отсутствие повторяемости результатов из-за разного порядка выполнения ассоциативных операций…

Локальность данных: множество открытых вопросов
(сложности описания алгоритмов)

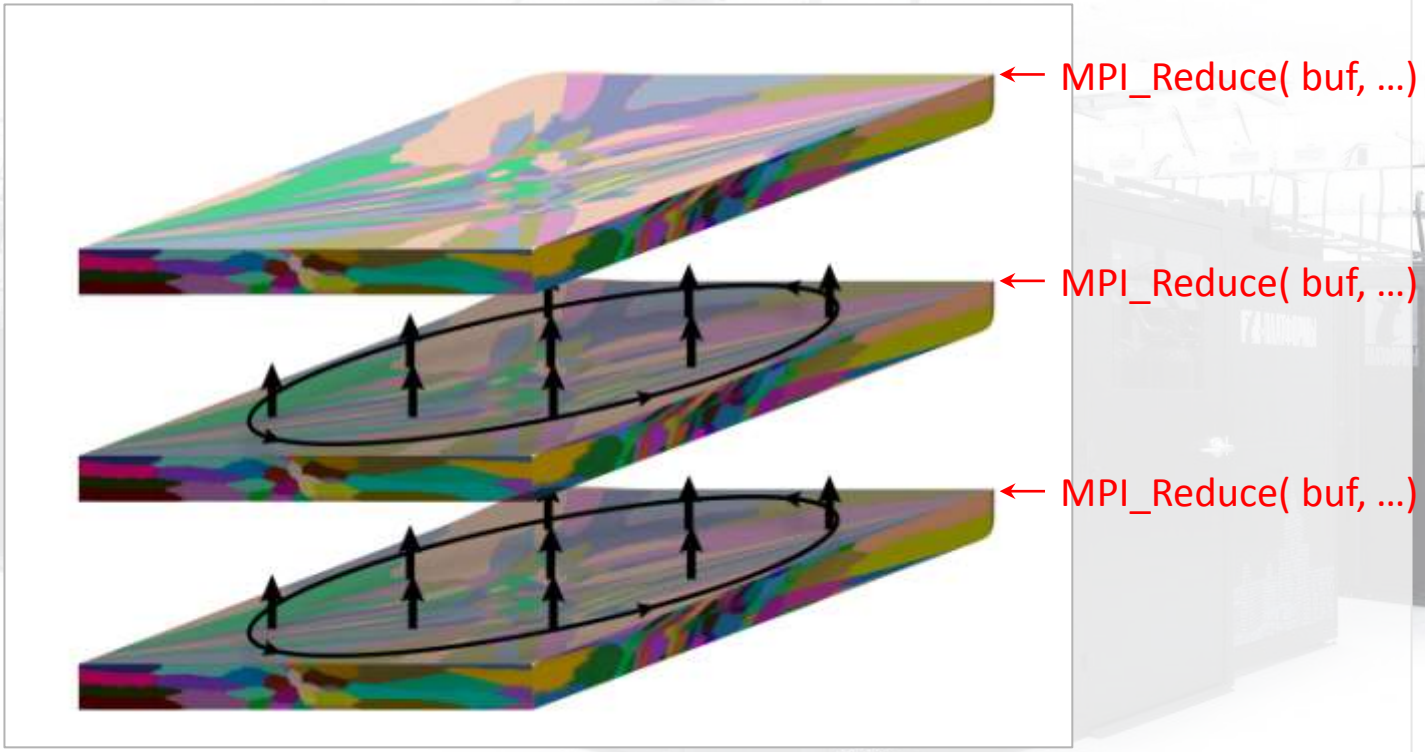
Как оценивать пространственную и временную локальность данных в программе ?

Как сравнивать пространственную и временную локальность данных разных программ ?

Можем ли мы предсказывать локальность данных в будущих реализациях, опираясь только на информацию об алгоритмах?

Что означает “локальность алгоритма” ?
Что означает “алгоритм обладает хорошей/плохой локальностью” ?
В алгоритмах нет структур данных, а значит и понятие “локальность” к ним напрямую не применимо! Вместе с этим, алгоритмы определяют суть программ, где локальность уместна и важна.

Коммуникационный профиль приложений
(сложности описания алгоритмов)



А что такое коммуникационный профиль приложений?
Как его получить и описывать?

Все ли мы знаем о разложении Холецкого ?

Да… Кажется, что Да…

Фундаментальный вопрос:

Что означает выполнить полное описание алгоритма?

Пока ответа не знает никто...

Разновидности одного алгоритма…
(сложности описания алгоритмов)

Файл Правка Вид Журнал Закладки Инструменты Справка

16th International Conference … | Метод Холецкого (нахо…

algowiki-project.org/ru/Метод_Холецкого_(нахождение_симметричного_треугольного_разло

Algowiki

Заглавная страница
Общий форум
Технический форум
Справка
Свежие правки

Хранилище файлов
Новые файлы
Загрузить файл

Инструменты
Ссылки сюда
Связанные правки
Спецстраницы
Версия для печати
Постоянная ссылка
Сведения о странице

На других языках
English

Метод Холецкого (нахождение симметричного треугольного разложения)

Основные авторы описания: И.Н.Коньшин

Содержание [убрать]

1 Разложение Холецкого (метод квадратного корня), базовый точечный вещественный вариант для плотной симметричной положительно определённой матрицы
- 1.1 -разложение
- 1.2 -разложение

2 Разложение Холецкого, блочный вещественный вариант для плотной симметричной положительно определённой матрицы

3 Разложение Холецкого, точечный вещественный вариант для разреженной симметричной положительно определённой матрицы
- 3.1 Основные отличия от случая плотной матрицы
- 3.2 Переупорядочивания для уменьшения количества новых ненулевых элементов

4 Разложение Холецкого, блочный вещественный вариант для разреженной симметричной положительно определённой матрицы

5 Разложение Холецкого для симметричной незнакоопределенной (седловой) матрицы

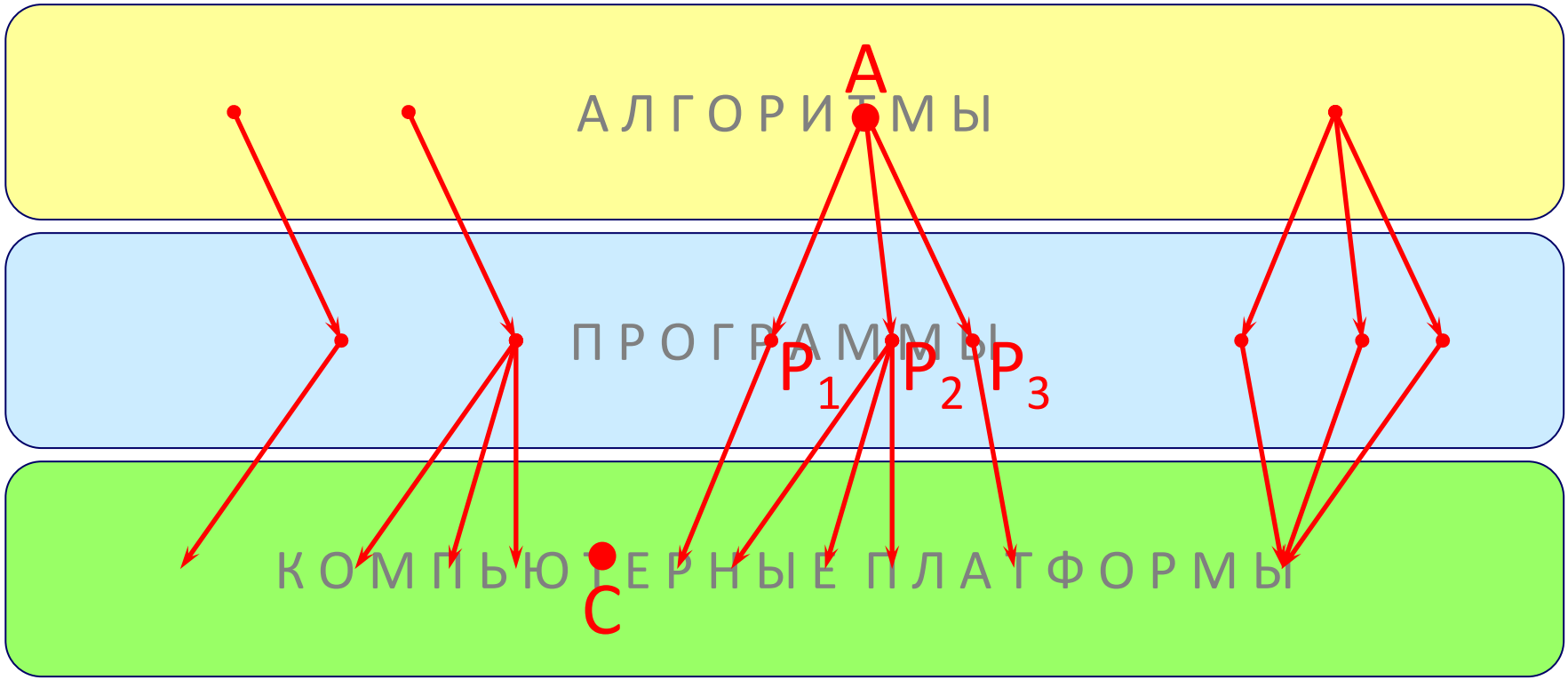
6 Разложение Холецкого для эрмитовой матрицы
- 6.1 Точечный вариант
- 6.2 Блочный вариант

7 Использование разложения Холецкого в итерационных методах
- 7.1 Ограничивание заполнения в разложении Холецкого
- 7.2 Неполное разложение Холецкого по позициям IC()
- 7.3 Приближенное разложение Холецкого по значениям IC()
- 7.4 Приближенное разложение Холецкого второго порядка IC()
- 7.5 Комбинация разложений Холецкого IC() и IC()

8 Использование разложения Холецкого в параллельных итерационных алгоритмах
- 8.1 Переупорядочивания для выделения блочности
- 8.2 Разложение в независимых блоках
- 8.3 Разложение в сепараторах
- 8.4 Иерархические и вложенные алгоритмы
- 8.5 Блочный метод Якоби
- 8.6 Аддитивный метод Шварца
- 8.7 Неполное обратное треугольное разложения

9 Решение линейных систем с треугольной матрицей
- 9.1 Решение системы с плотной нижнетреугольной матрицей

Алгоритмы и их реализации



Достаточно ли у нас информации об алгоритме A, чтобы создать эффективную реализацию для платформы C?

Можем ли мы использовать реализации P_1, P_2, P_3 или мы должны создавать еще одну реализацию P для платформы C?

Структура и свойства алгоритмов
(от мобильных платформ до экзафлопсных суперкомпьютеров)

Алгоритмы:
теоретический потенциал
(машинно-независимые свойства)

Алгоритмы:
особенности реализации

AlgoWiki

http://AlgoWiki-Project.org

AlgoWiki

Файл Правка Вид Журнал Закладки Инструменты Справка

Алговики

algowiki-project.org/ru/Открытая_энциклопедия_свойств_алгоритмов

Поиск

Войти Запрос учётной записи

Main page Обсуждение Читать Просмотр История Поиск

Algowiki

Заглавная страница
Общий форум
Технический форум
Справка
Свежие правки

Хранилище файлов
Новые файлы
Загрузить файл

Инструменты
Ссылки сюда
Связанные правки
Спецстраницы
Версия для печати
Постоянная ссылка
Сведения о странице

На других языках
English

Открытая энциклопедия свойств алгоритмов

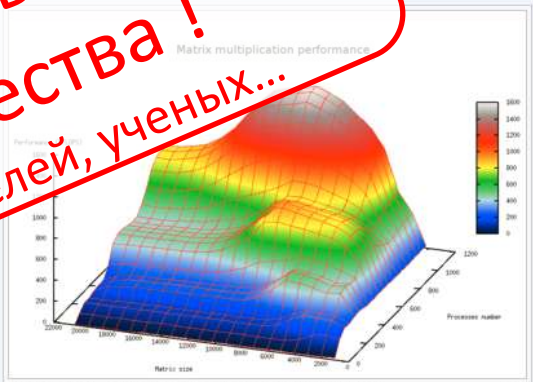
Добро пожаловать! Присоединяйтесь!

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 позволяет оценить его потенциал применительно к конкретной параллельной вычислительной платформе. Кроме классических свойств алгоритмов, например, последовательной сложности, в AlgoWiki представлены полезные сведения, составляющие в совокупности полную картину об алгоритме: параллельная сложность, параллельная структура, детерминированность, локальности данных, эффективность и масштабируемость, коммуникационный профиль, известных реализаций и многое другое.

Читать подробнее о проекте

Изображение дня



Производительность умножения плотных матриц

Структура проекта

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

Организация работы

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

Образцовая статья

Разложение Холецкого (метод квадратного корня)

1 Свойства и структура алгоритма

1.1 Общее описание алгоритма

Свойства алгоритма:

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

Участники проекта

AlgoWiki – это проект для всего вычислительного сообщества !

Для студентов, аспирантов, преподавателей, ученых…

http://AlgoWiki-Project.org

CALL FOR PAPERS

"Supercomputing Co-Design Technology"

(http://agora.guru.ru/SCDT)

As part of **ICA3PP 2016**
16th International Conference on Algorithms and Architectures for Parallel Processing
(http://www.arcos.inf.uc3m.es/wp/ica3pp2016)

14–15th DECEMBER 2016, GRANADA, SPAIN

SCOPE AN

The computing world is changing rapidly. All de
high-performance supercomputers – are b

IMPORTANT DATES

Paper Submission Deadline: July 25th, 2016

Author Notification: September 15th, 2016

Camera-Ready Paper Due: October 15th, 2016

Workshop Dates: December 14th–15th, 2016

PROGRAM COMMITTEE

- **Victor Gergel, co-chair**, Lobachevsky State University of Nizhni Novgorod, Russia
- **Vladimir Voevodin, co-chair**, Moscow State University, Russia
- Arndt Bode, Leibniz Supercomputing Centre, Germany
- Alexander Boukhanovsky, ITMO University, Russia
- Yuefan Deng, Stony Brook University, USA
- Florent de Dinechin, INSA Lyon, France
- Torsten Hoefler, Swiss Federal Institute of Technology, Switzerland
- Thomas Ludwig, German Climate Computing Center, Germany
- Iosif Meyerov, Lobachevsky State University of Nizhni Novgorod, Russia
- Marek Michalewicz, A*STAR Computational Resource Centre, Singapore
- Bernd Mohr, Jülich Supercomputing Centre, Germany
- Mikhail Moshkov, King Abdullah University of Science and Technology, Saudi Arabia
- Sergey Orlov, Transport and Telecommunication Institute, Latvia
- Nina Popova, Moscow State University, Russia
- Arnold Rosenberg, Northeastern University, USA
- Ahmed Seffah, Lappeenranta University of Technology, Finland
- Yaroslav Sergeyev, University of Calabria, Italy
- Andrey Sozykin, Ural Federal University, Russia
- Roman Wyrzykowski, Czestochowa University of Technology, Poland

PUBLICATIONS

All papers selected by the Program Committee and presented at the Workshop will be published in **the Workshop Proceedings by Springer LNCS series**. After the conference, the authors of the papers presented at the workshop will be invited to submit extended versions of the papers in the following scientific journal: Supercomputing Frontiers and Innovations (http://superfri.org/superfri).

International Conference

Russian Supercomputing Days

Москва
26-27 сентября 2016 г.

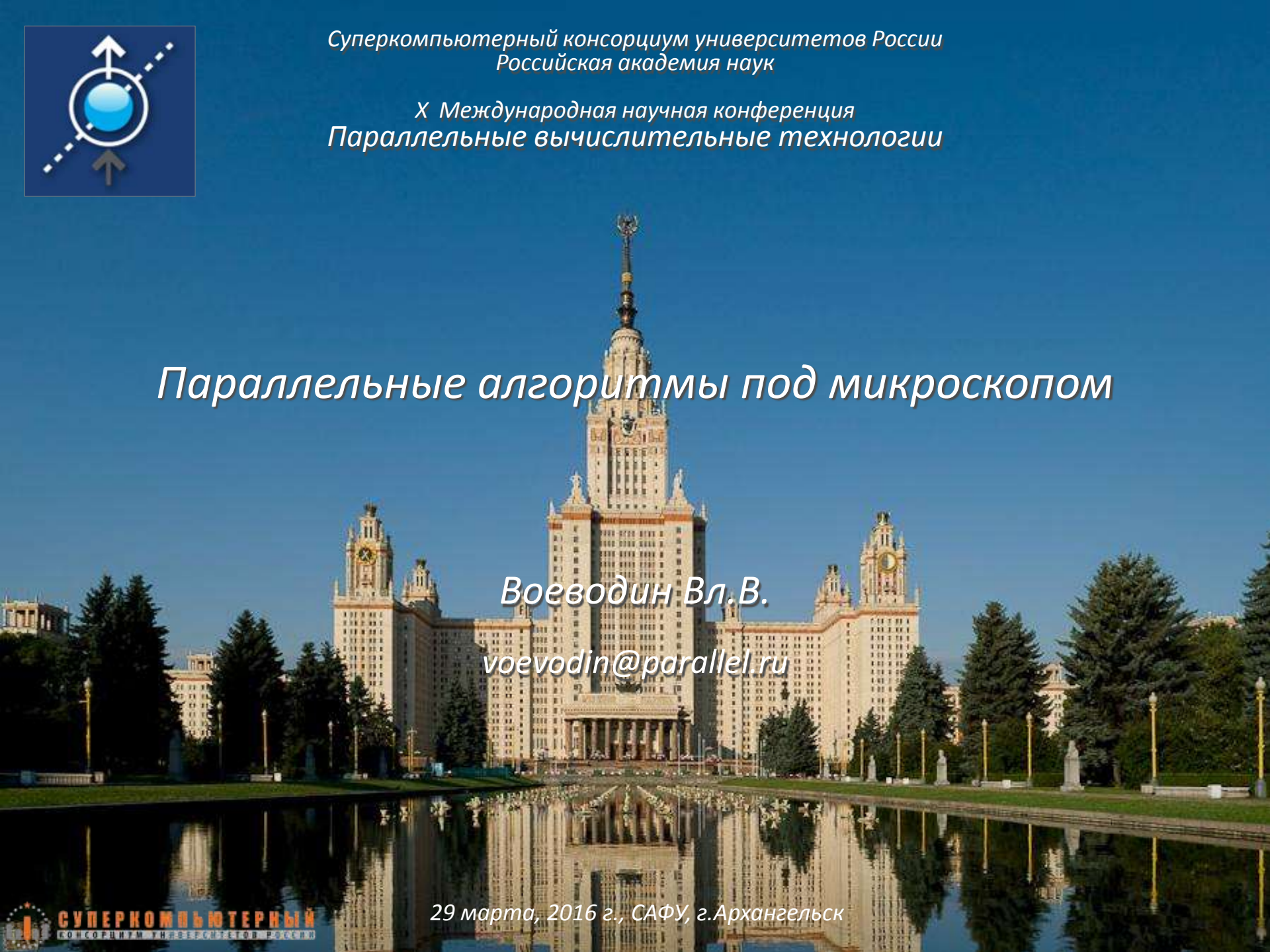
«Суперкомпьютерные дни в России» — новая серия ежегодных международных суперкомпьютерных конференций, в основу которой легли давние традиции российских и международных суперкомпьютерных мероприятий.

RussianSCDays.org

СУПЕРКОМПЬЮТЕРНЫЕ ДНИ В РОССИИ 2015

- Два исключительно наполненных суперкомпьютерными событиями дня
- 450 участников из России, Европы, Китая, Японии, США и других стран
- Представители 35 университетов, 28 институтов РАН, 60 коммерческих компаний
- Пленарные доклады и 7 параллельных секций
- Стендовая секция
- Суперкомпьютерная выставка
- Конференция молодых ученых
- Тренинги от ведущих HPC-компаний: Intel, Mellanox, NVIDIA, HP, PCK ...
- Анонс рейтинга Top50, индустриальные секции, тренинги для школьных учителей информатики, дискуссионные группы, 3D-визуализация и т.п.

- 1 апреля 2016 г. - начало приема докладов
- 1 июня - представление аннотаций докладов
- 15 июня - представление полных версий докладов
- 15 июля - уведомление о включении доклада в программу конференции
- 30 июля - представление окончательного варианта статьи



Суперкомпьютерный консорциум университетов России
Российская академия наук

X Международная научная конференция
Параллельные вычислительные технологии

Параллельные алгоритмы под микроскопом

Воеводин Вл.В.

voevodin@parallel.ru

29 марта, 2016 г., САФУ, г.Архангельск

СУПЕРКОМПЬЮТЕРНЫЙ
КОНСОРЦИУМ УНИВЕРСИТЕТОВ РОССИИ