

**Исследование ускорения  
вычислений параллельных  
реализаций метода конечных  
элементов для уравнений  
мелкой воды**

Дементьева Екатерина

# Цели работы

- Подготовка входных данных о сетках и триангуляции для расчетов на многопроцессорной ВС.
- Анализ ускорения и эффективности распараллеливания метода конечных элементов для решения краевой задачи для уравнений мелкой воды с помощью библиотеки MPI на многопроцессорной ВС.

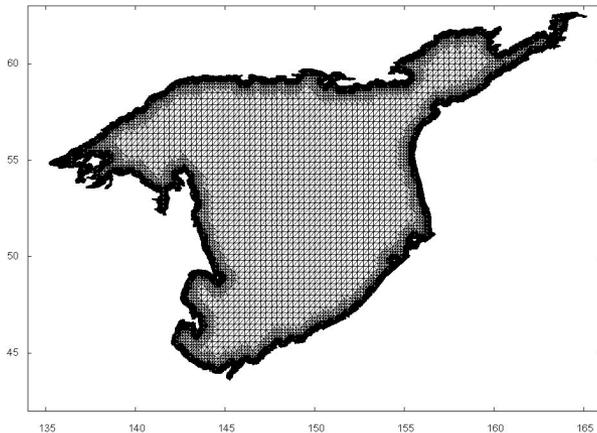
# В данной работе:

- было проведено исследование эффективности распараллеливания метода конечных элементов для решения краевой задачи для уравнений мелкой воды с помощью библиотеки MPI на многопроцессорной ВС. Был выполнен анализ производительности нескольких параллельных реализаций алгоритма численного решения задачи;
- реализованы два алгоритма декомпозиции прямоугольной вычислительной области с триангуляцией для равномерного распределения вычислительной нагрузки по процессорам.

# Дифференциальная постановка прямой задачи

$(r, \lambda, \theta)$  — сферическая система координат с началом в центре земного шара.

$$\varphi = \theta + \pi/2 \quad 0 \leq \varphi < \pi \quad 0 \leq \lambda \leq 2\pi$$



$$\begin{aligned} \frac{\partial u}{\partial t} &= lv + mg \frac{\partial \xi}{\partial \lambda} - R_f u + f_1 \\ \frac{\partial v}{\partial t} &= -lu + ng \frac{\partial \xi}{\partial \varphi} - R_f v + f_2 \\ \frac{\partial \xi}{\partial t} &= m \left( \frac{\partial}{\partial \lambda} (Hu) + \frac{\partial}{\partial \varphi} \left( \frac{n}{m} H v \right) \right) + f_3 \end{aligned} \quad (1)$$

$R_f = r_* |\mathbf{U}| / H$  — сила трения  $r_*$  — коэффициент трения;

$l = -2\omega \cos \varphi$  — параметр Кориолиса;  $m = 1/(R_E \sin \varphi)$ ;  $n = 1/R_E$ ;  
 $g$  — ускорение силы тяжести

Уравнения (1) – уравнения мелкой воды.

$$HU_n + \beta m_2 \sqrt{gH} \xi = m_2 \sqrt{gH} d \quad \text{на } \Gamma \times (0, T) \quad (2)$$

$U_n = \mathbf{U} \cdot \mathbf{n}$ ,  $\mathbf{n} = (n_1, \frac{n}{m} n_2)$  — вектор внешней нормали к границе;

$\beta \in [0, 1]$  — заданный параметр;

$$u(0, \lambda, \varphi) = u_0(\lambda, \varphi), \quad v(0, \lambda, \varphi) = v_0(\lambda, \varphi), \quad \xi(0, \lambda, \varphi) = \xi_0(\lambda, \varphi)$$

Постановка дифференциальной задачи выполнена В.И. Агошковым.

Для дискретного аналога задачи В.В.Шайдуровым и Е.Д. Кареповой получены априорные оценки устойчивости, а также показан второй порядок аппроксимации во внутренних узлах равномерной сетки.

# Векторно-матричная форма дискретного аналога. Потенциальный параллелизм

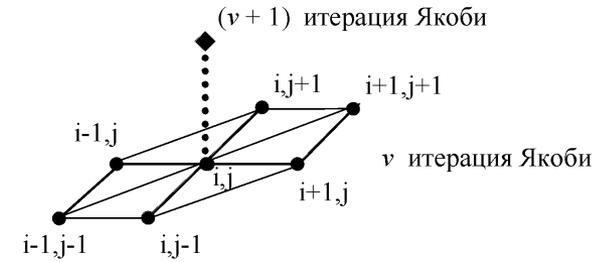
Для фиксированного момента времени  $t^{k+1}$  найти вектор  $\mathbf{V}^{k+1}$ , удовлетворяющий системе линейных алгебраических уравнений

$$\mathbf{A}^{k+1} \mathbf{V}^{k+1} = \mathbf{F}^{k+1}, \quad (21)$$

где

$$\begin{aligned} \mathbf{V}^{k+1} &= (u_1, \dots, u_{N_{nd}}, v_1, \dots, v_{N_{nd}}, \xi_1, \dots, \xi_{N_{nd}}), \\ \mathbf{F}^{k+1} &= (f_1^h(\Psi_1), \dots, f_1^h(\Psi_{N_{nd}}), f_2^h(\Psi_1), \dots, f_2^h(\Psi_{N_{nd}}), \\ &\quad f_3^h(\Psi_1) + b^h(d_1^h, \Psi_1), \dots, f_3^h(\Psi_{N_{nd}}) + b^h(d_{N_{nd}}^h, \Psi_{N_{nd}})). \\ \mathbf{V}^{(l+1)} &= \mathbf{V}^{(l)} - \mathbf{D}^{-1} (\mathbf{A} \mathbf{V}^{(l)} - \mathbf{F}) \end{aligned}$$

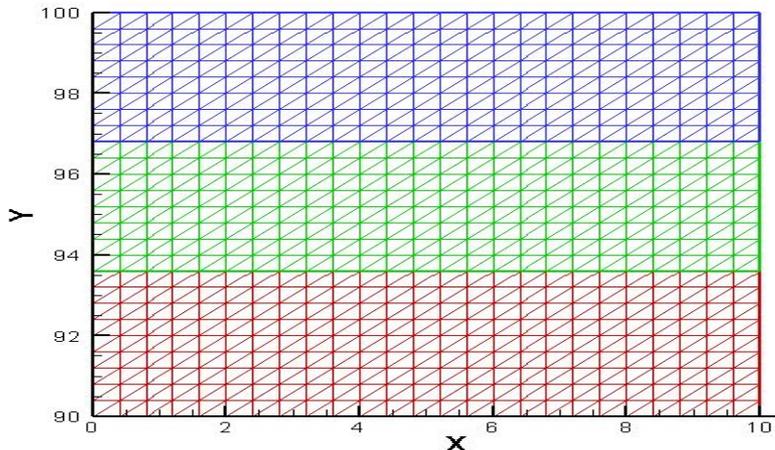
# Декомпозиция области



## Без теневых граней

Исходная область **не включает** взаимно перекрывающиеся подобласти.

Пересчет значений на границах между подобластями предполагает **обмен с дополнительным суммированием** при обмене на каждой итерации Якоби

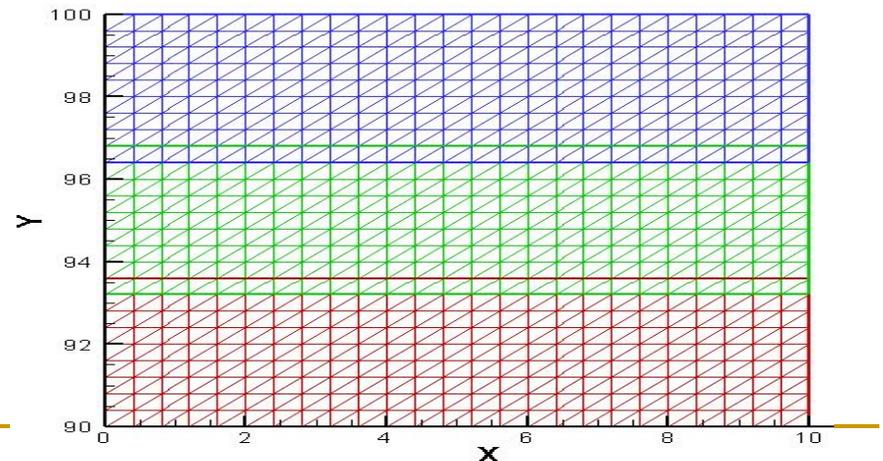


## С теневыми гранями

Исходная область **включает** взаимно перекрывающиеся подобласти.

Невязка в итерации Якоби в граничных точках  $i$ -го процесса **насчитывается в подобласти соседних** процессов.

Семиточечный шаблон предполагает перекрытие в два слоя расчетных точек.



# Однородное распределение данных по компьютерам

## Баланса времени

- затрачиваемого на вычисления
- затрачиваемого на взаимодействия ветвей параллельной программы

## Требуемый результат

- равенство объёмов распределяемых частей данных

# Программа распределения данных по процессам

- Написана на языке программирования C
- **На входе**
  - файл, описывающий сетку координат с батиметрией
  - файл, описывающий триангуляцию
- **На выходе**
  - файлы, подготовленные для каждого процесса, описывающие декомпозицию (содержат информацию об общих данных и теневых гранях) и соответствующую часть сетки координат с батиметрией
  - файлы триангуляции , также подготовленные для каждого процесса

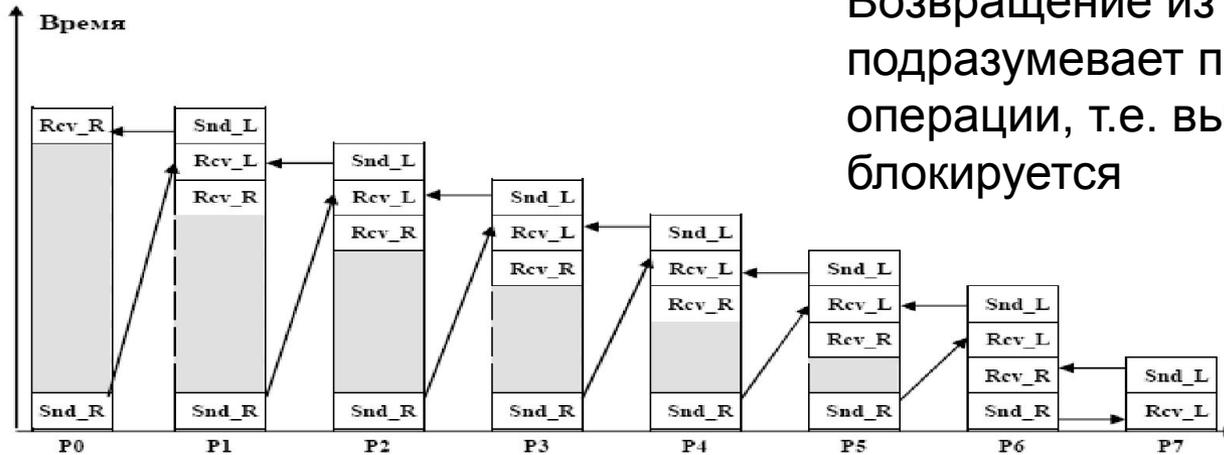
Отметим, что поскольку декомпозиция с теневыми гранями на  $P$  процессов требует дополнительного хранения в каждой граничной точке подобласти семь коэффициентов матрицы жесткости, три значения вектора решения текущей и предыдущей итерации и значение правой части, то необходимо на

$$28(P-1)N_{-bnd} * \text{SizeOfDouble} \text{ байт}$$

больше, чем для декомпозиции без перекрытий.

# Типы обменов. Блокирующие передачи

Возвращение из функции подразумевает полное окончание операции, т.е. вызывающий процесс блокируется



Реализации обменов по цепочке процессов с помощью функций совмещенных приема-передачи `MPI_Sendrecv(...)`.

- Все процессы кроме последнего отправляют данные своим правым соседям и от них же ожидают поступления данных.
- Все, кроме первого процесса, посылают данные своим левым соседям и ожидают поступления данных от них же.

---

# Типы обменов. Неблокирующие передачи

*Неблокирующие функции* подразумевают совмещение операций обмена с другими операциями

Время, затрачиваемое на обмены с использованием неблокирующих передач не зависят от количества участвующих в обменах процессов

---

---

# Численные эксперименты

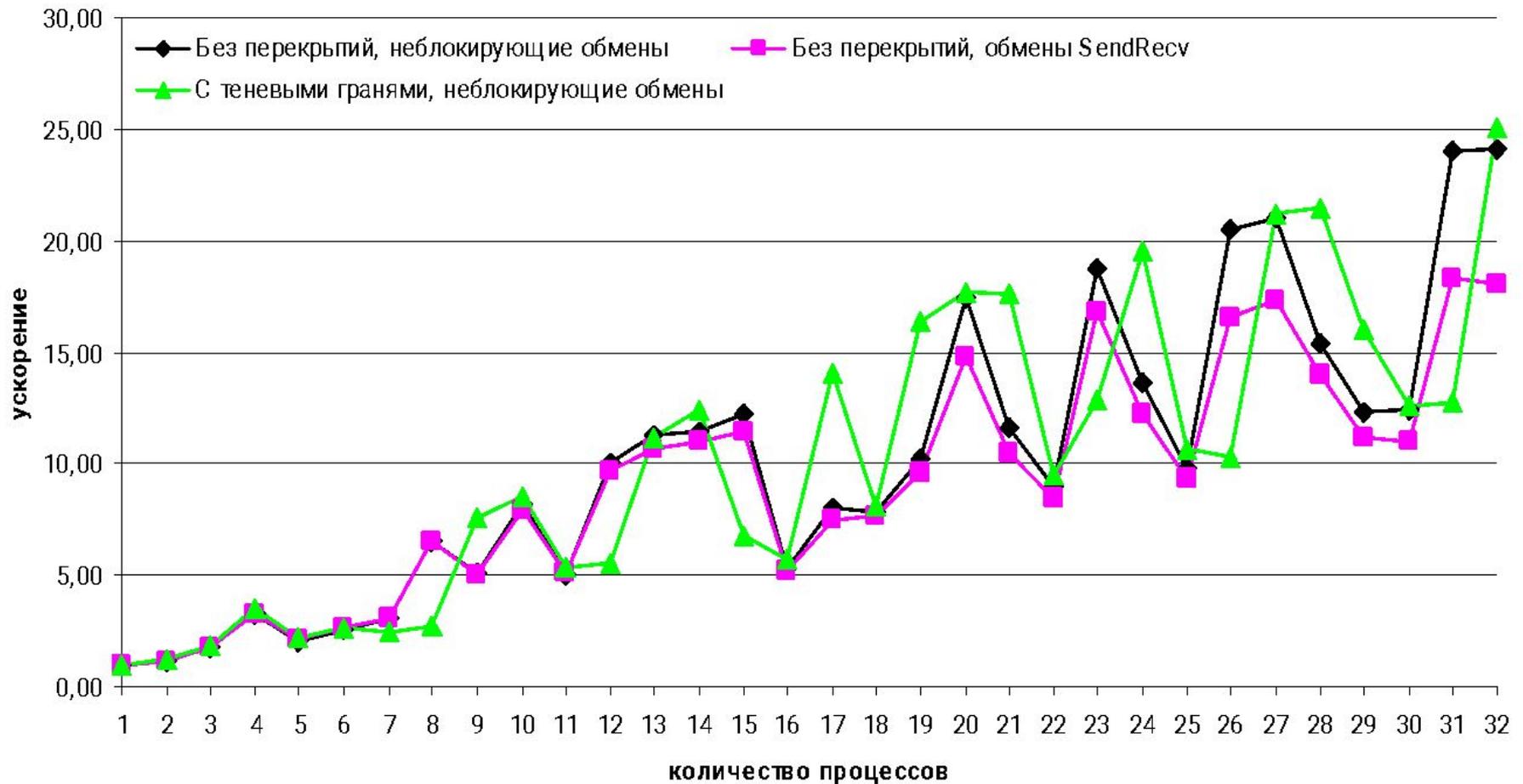
---

# Кластер МВС-1000/ИВМ неоднородной архитектуры (собственная сборка ИВМ СО РАН)

## **99** вычислительных ядра

- 23 вычислительных узла AMD Athlon64/3500+/1Gb (однопроцессорные, одноядерные);
- 12 вычислительных узлов AMD Athlon64 X2 Dual Core/4800+/2Gb (однопроцессорные, двухъядерные);
- 12 вычислительных узлов AMD Athlon64 X2 Dual Operon 2216МГц/4Gb (двупроцессорные, двухъядерные);  
управляющий узел, сервер доступа и файловый сервер Athlon64/3500+/1Gb с общей дисковой памятью 400 Гб;
- управляющая сеть кластера - FastEthernet (100 Мбит/сек);
- сеть передачи данных - GigaEthernet (1000 Мбит/сек).

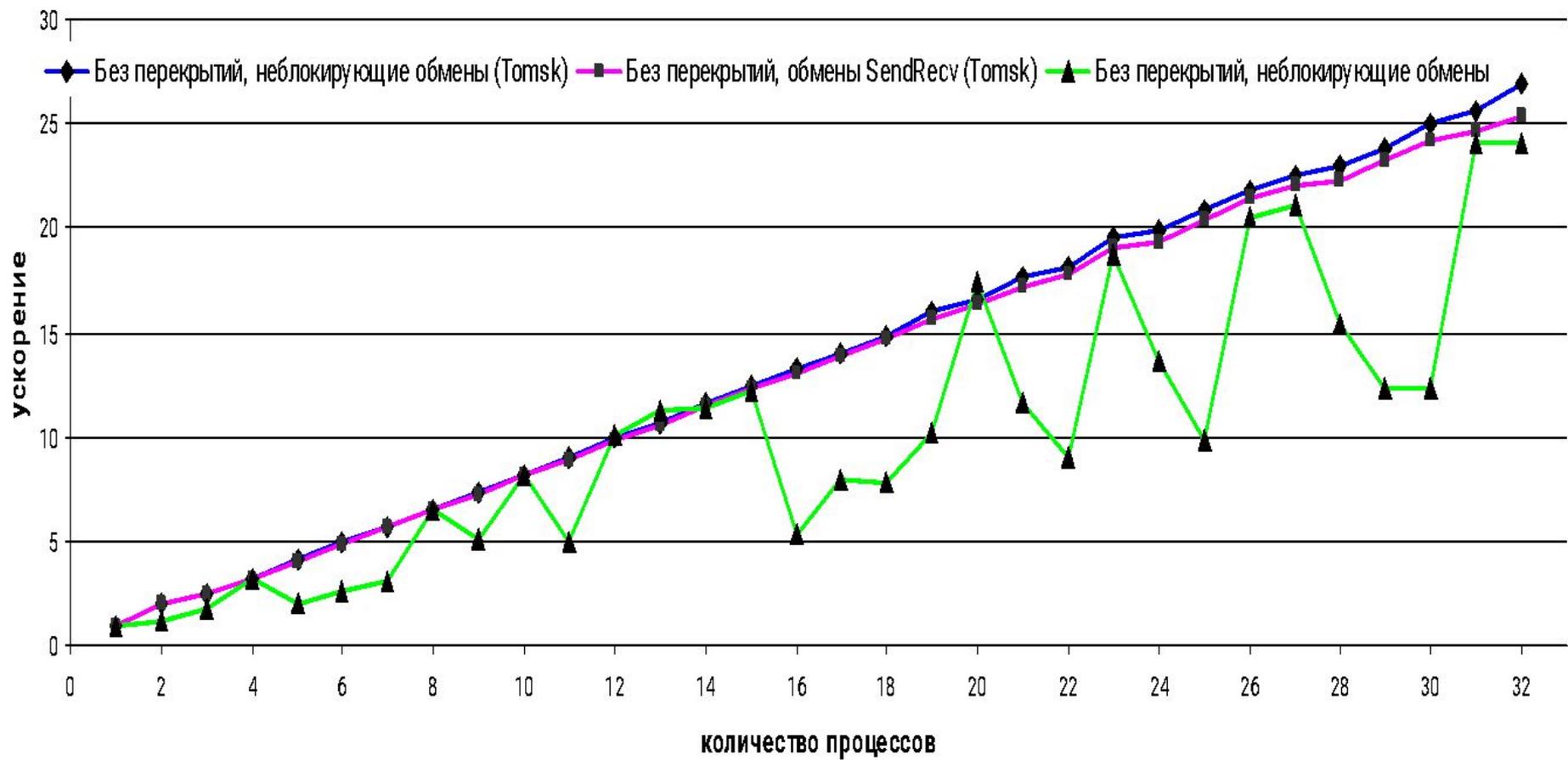
# Задача с точным решением на модельной сетке 801 x 801



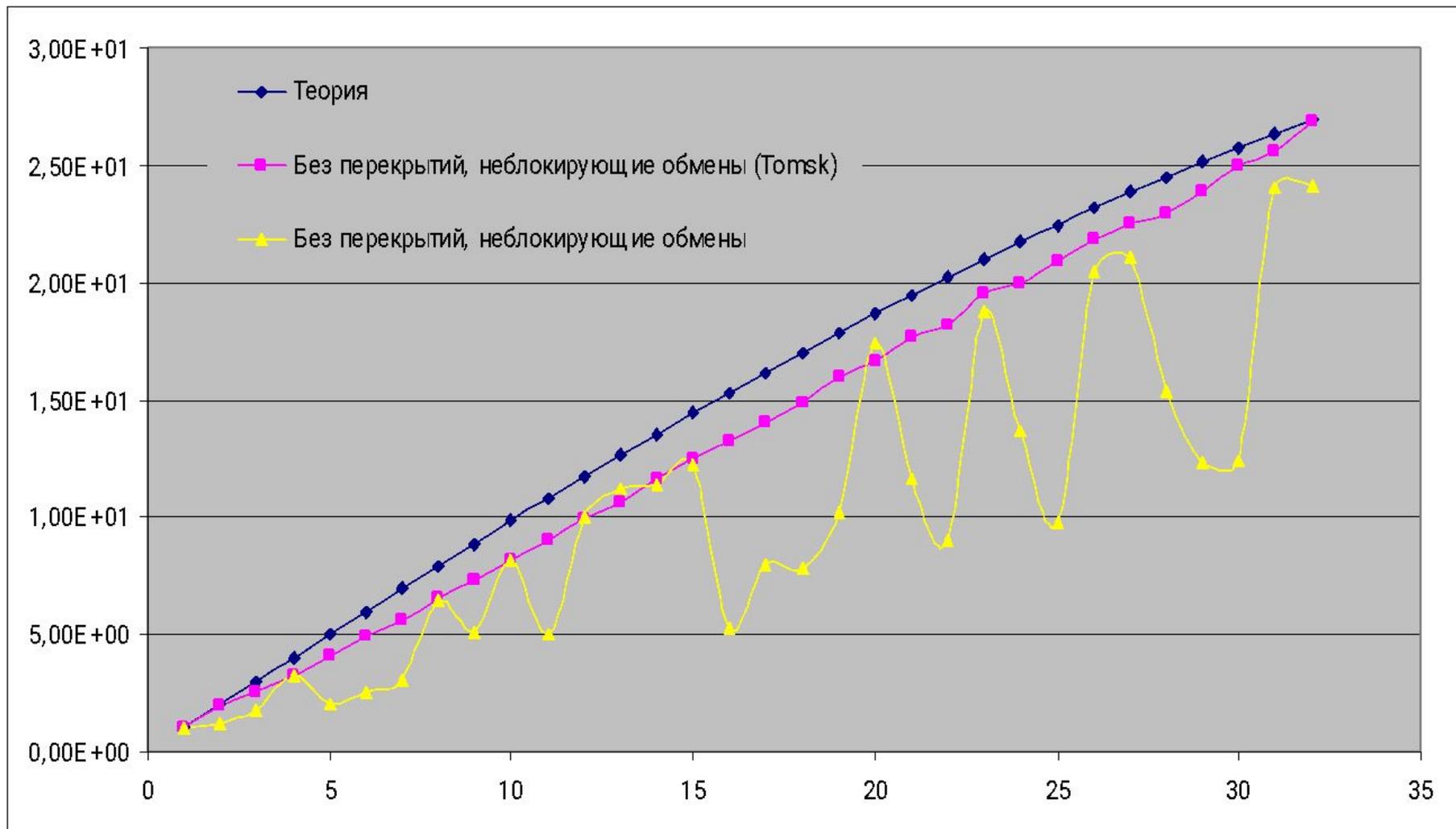
# Кластер Skif Cyberia (ТГУ)

- Архитектура x86 с поддержкой 64 разрядных расширений. Количество вычислительных узлов/процессоров 283/566 (один узел - управляющий) (**1132 ядра**)
- Тип процессора: двухъядерный Intel®Xeon® 5150, 2,66ГГц (Woodcrest)
- Скорость передачи сообщений между узлами 950 Мб/сек с задержкой не более 2,5 мкс
- Пиковая производительность 12 Тфлопс
- Реальная производительность на тесте Linpack 9,013 Тфлопс (75% от пиковой)
- Суммарный объем оперативной памяти 1 136 Гб
- Суммарный объем дискового пространства 22,56 Тб Внешняя дисковая система хранения данных 10 Тб Параллельная файловая система суммарная пропускная способность 700 Мб/сек
- Потребляемая мощность 90 кВт

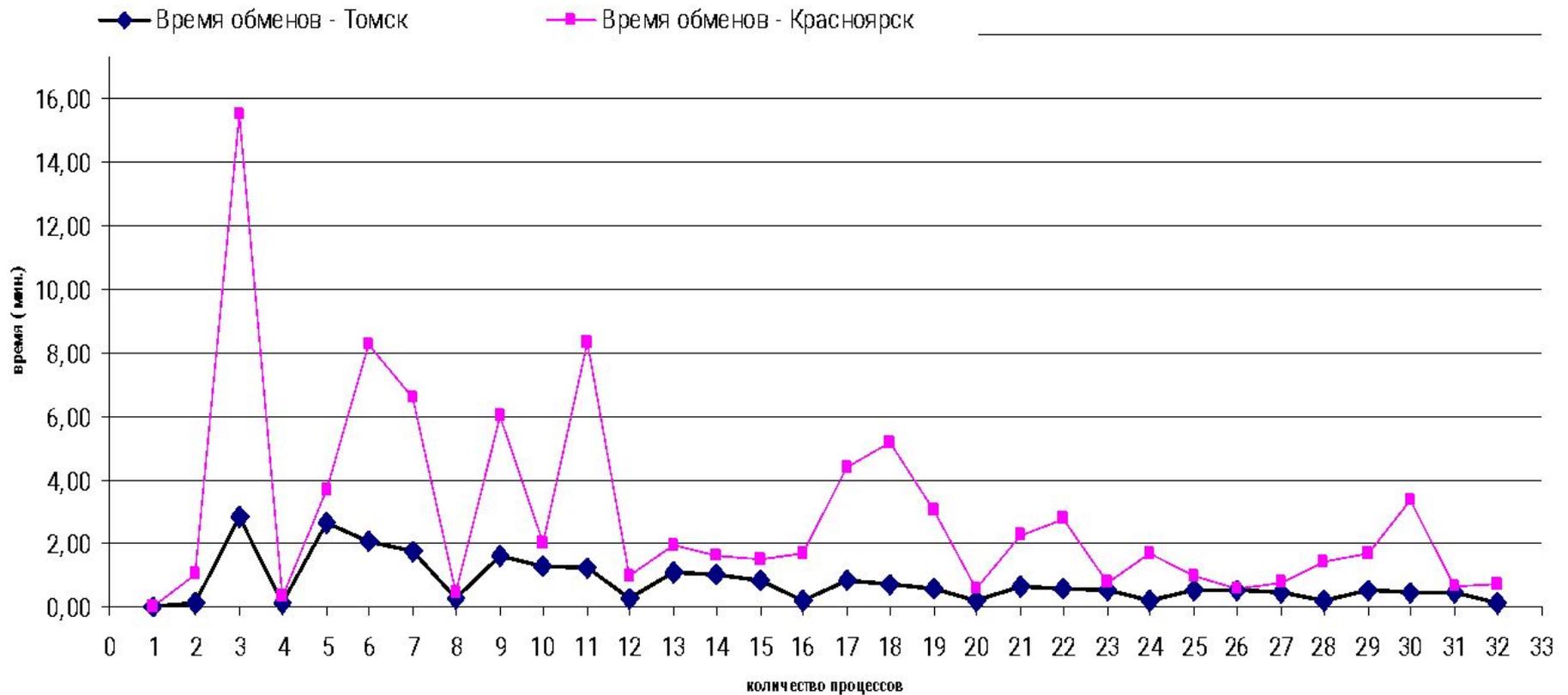




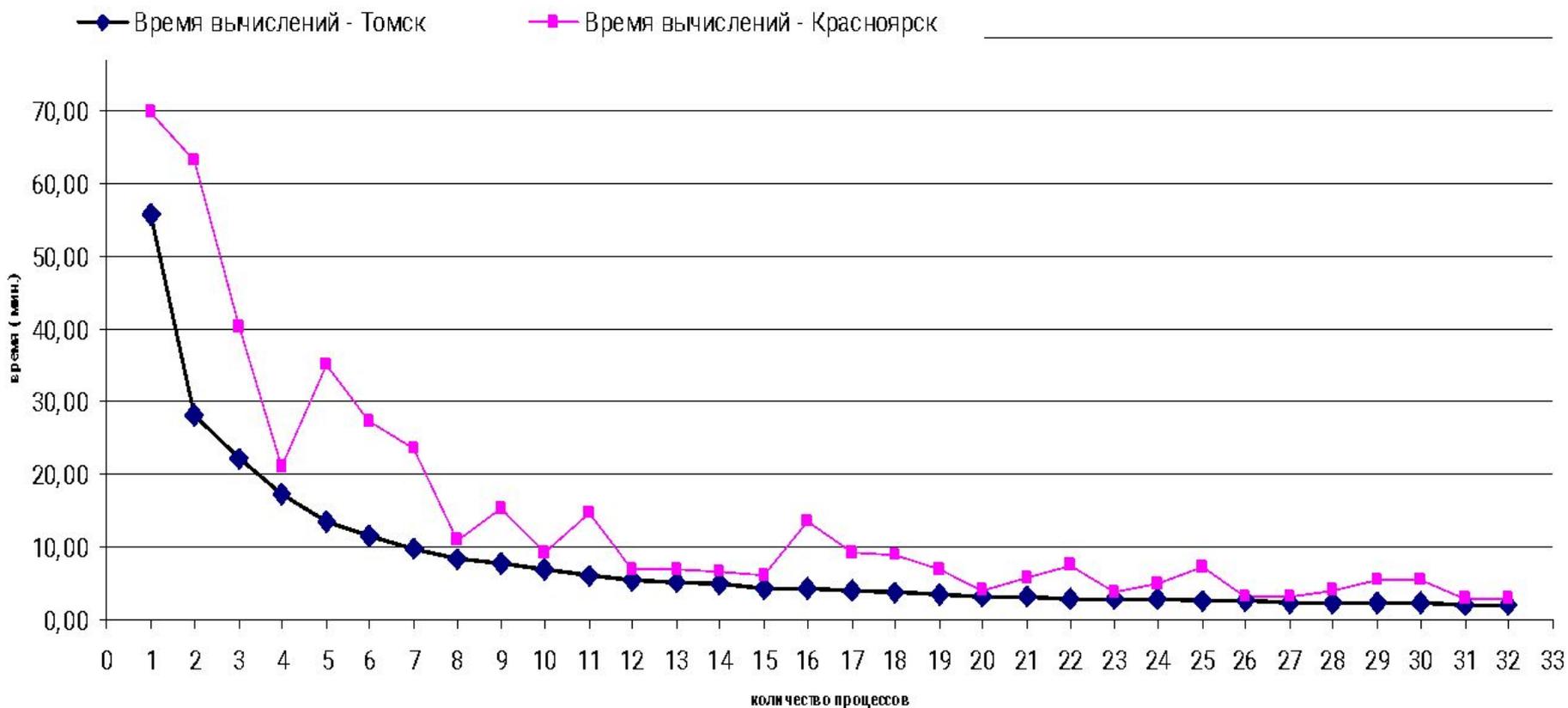
# Сравнение результатов с теоретическими оценками



# Время неблокирующих обменов 801x801 без перекрытий



# Время вычислений 801x801 без перекрытий



---

## Результаты исследования показали, что

- ускорение вычислений при увеличении количества процессов для сеток больших размерностей не убывает и близко к линейному. Вычислительные эксперименты по изучению ускорения демонстрируют хорошее согласование с теоретическими оценками
  - поведение ускорения сильно зависит от особенностей архитектуры используемой вычислительной системы. Неоднородность архитектуры может влиять на величину ускорения не лучшим образом и приводить к необоснованно большому времени, затрачиваемому на вычисления при использовании нескольких процессов
-

# В результате:

- проведено исследование эффективности распараллеливания метода конечных элементов для решения краевой задачи для уравнений мелкой воды;
- реализованы два алгоритма декомпозиции прямоугольной вычислительной области с триангуляцией для равномерного распределения вычислительной нагрузки по процессорам;
- написаны shell-скрипты, позволяющие запускать серию вычислительных экспериментов под ОС Linux, собирать данные о времени выполнения в файлы для последующей обработки;
- создана программная оболочка Microsoft Excel, с помощью которой можно легко и удобно анализировать полученные данные.