Решение дифф. уравнений на CUDA на примере задач аэро-гидродинамики.

- Лектор:
  - Сахарных Н.А. (ВМиК МГУ, NVidia)

План

- Постановка задачи
- Численный метод
- Обзор архитектуры GPU и модели CUDA
- Особенности реализации
- Результаты и выводы
- Полезные ссылки

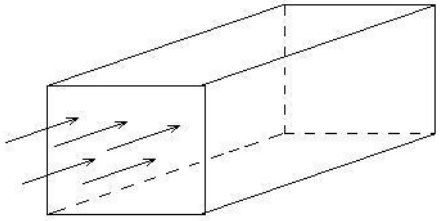
Введение

- Вычислительные задачи аэро-гидродинамики
  - Моделирование турбулентных течений

- ВМиК МГУ, кафедра мат. физики
  - Пасконов В.М., Березин С.Б.

Постановка задачи

- Течение вязкой несжимаемой жидкости в 3D канале

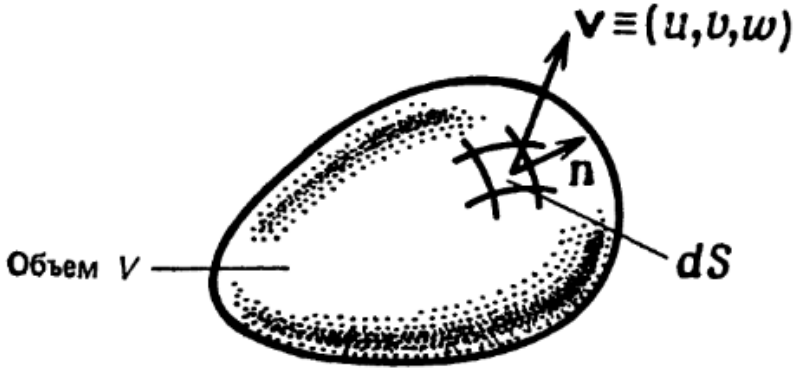


  - Канал заполнен и находится в однородной среде
  - Произвольные начальные и граничные условия
  - Неизвестные величины – скорость и температура

Основные уравнения

- Полная система уравнений Навье-Стокса в безразмерных величинах
  - Уравнение неразрывности
  - Уравнения движения (Навье-Стокса)
  - Уравнение энергии

Уравнение неразрывности



$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

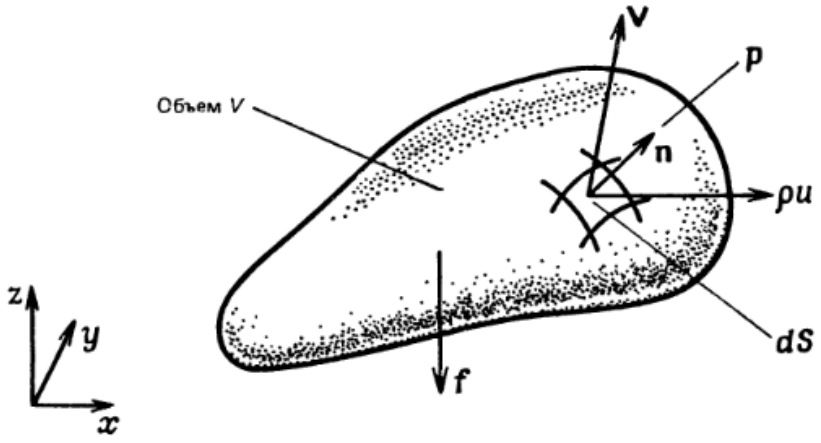
$$\rho = const$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

- Используется при выводе остальных уравнений (движения и энергии)
- Проверка точности текущего решения

Уравнения Навье-Стокса

- Второй закон Ньютона: $\int_V \rho \frac{D\mathbf{v}}{Dt} dV = \sum \mathrm{F}$



Невязкая жидкость:

$$\sum \mathrm{F} = \int_V (\rho \mathbf{f} - \nabla p) dV$$

Вязкая жидкость:

$$\sum \mathrm{F} = \int_V (\rho \mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\tau}) dV$$

$\mathbf{f}$ – массовые силы (сила тяжести)

$\boldsymbol{\tau}$ – тензор вязких напряжений

p – давление

Безразмерные уравнения

- Параметры подобия
  - Число Рейнольдса $\mathrm{Re} = \dfrac{V' L'}{\mu'}$
  - Число Прандтля $\mathrm{Pr} = \dfrac{\mu' c_p'}{k'}$

V', L' – характерная скорость, размер

μ' – динамическая вязкость среды

k' – коэффициент теплопроводности

c_p' – удельная теплоемкость

- Уравнение состояния для идеального газа/жидкости:

$$p = \rho R T$$

Уравнения движения

- Безразмерная форма:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{\partial T}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right),$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{\partial T}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right),$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{\partial T}{\partial z}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right),$$

  - Уравнение состояния $p=T$

Уравнение энергии

- Первый закон термодинамики для объема V:

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}+w\frac{\partial T}{\partial z}=\frac{1}{\mathrm{Pr}\cdot\mathrm{Re}}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)+\frac{\gamma-1}{\gamma}\frac{1}{\mathrm{Re}}\Phi$$

- Диссипативная функция:

$$\Phi=\Phi_x+\Phi_y+\Phi_z$$

$$\Phi_x=2\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial w}{\partial x}\right)^2+\frac{\partial v}{\partial x}\frac{\partial u}{\partial y}+\frac{\partial w}{\partial x}\frac{\partial u}{\partial z}$$

$$\Phi_y=\left(\frac{\partial u}{\partial y}\right)^2+2\left(\frac{\partial v}{\partial y}\right)^2+\left(\frac{\partial w}{\partial y}\right)^2+\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}+\frac{\partial w}{\partial y}\frac{\partial v}{\partial z}$$

$$\Phi_z=\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2+2\left(\frac{\partial w}{\partial z}\right)^2+\frac{\partial u}{\partial z}\frac{\partial w}{\partial x}+\frac{\partial v}{\partial z}\frac{\partial w}{\partial y}$$

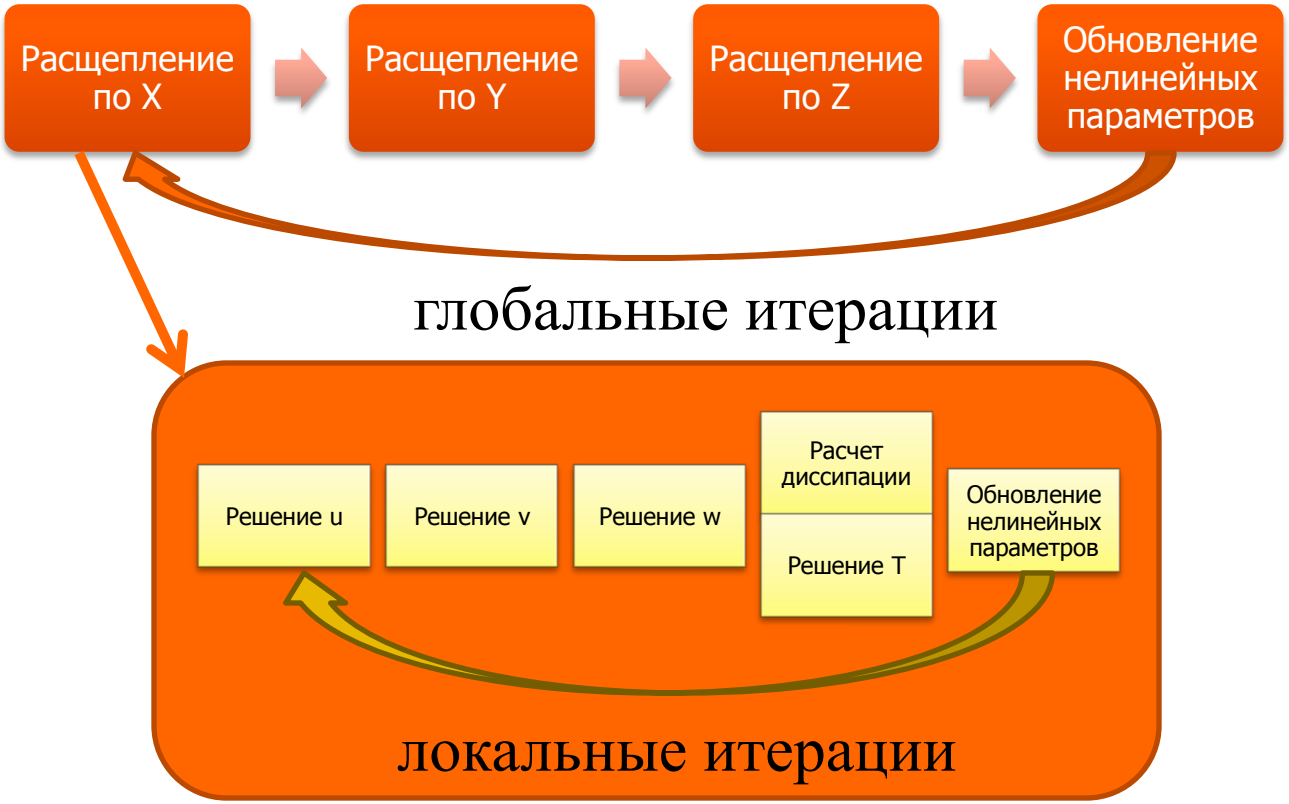
Финальные уравнения

- 4 нелинейных уравнения
  - НС + энергия
- Неизвестные величины:
  - Компоненты скорости: u, v, w
  - Температура: T

Численный метод

- Метод по-координатного расщепления
- Неявная схема 2-го порядка
- Нелинейные итерации
  - Полусумма значений на двух предыдущих итерациях

Численный метод



Стадии алгоритма

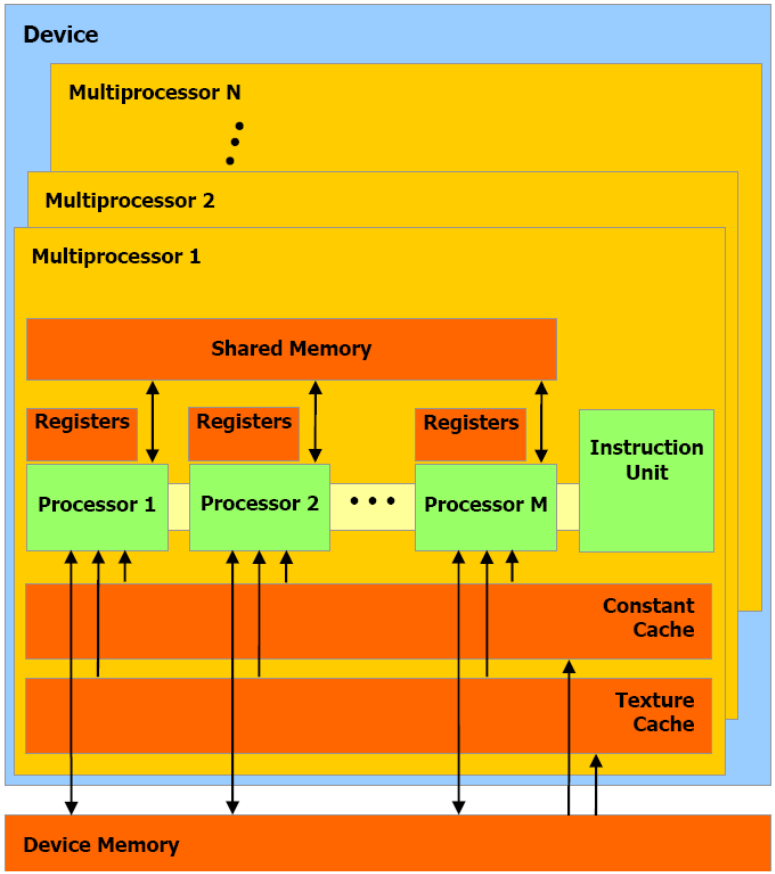
- Решение большого количества трехдиагональных СЛАУ
- Вычисление диссипации в каждой ячейке сетки
- Обновление нелинейных параметров

Особенности метода

- Большой объем обрабатываемых данных
- Высокая арифметическая интенсивность
- Легко параллелится

Архитектура GPU

- Массивный параллеллизм потоков
- Широкая пропускная способность памяти
- Поддержка двойной точности

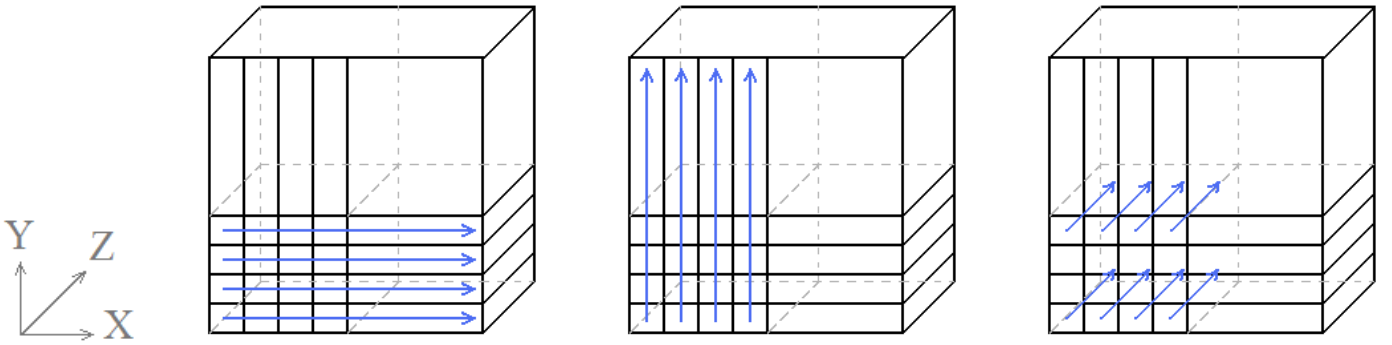


Реализация на CUDA

- Все данные хранятся в памяти GPU
  - 4 скалярных 3D массива для каждой переменной (u, v, w, T)
  - 3 дополнительных 3D массива
- ~300MB для сетки 128^3 в double

Решение трехдиагональных СЛАУ

- Каждая нить решает ровно одну трехдиагональную СЛАУ
  - На каждом шаге N^2 независимых систем



Расщепление X Расщепление Y Расщепление Z

Метод прогонки

- Необходимо 2 дополнительных массива
  - хранение: локальная память
- Прямой ход
  - вычисление a[i], b[i]
- Обратный ход
  - x[i] = a[i+1] * x[i+1] + b[i+1]

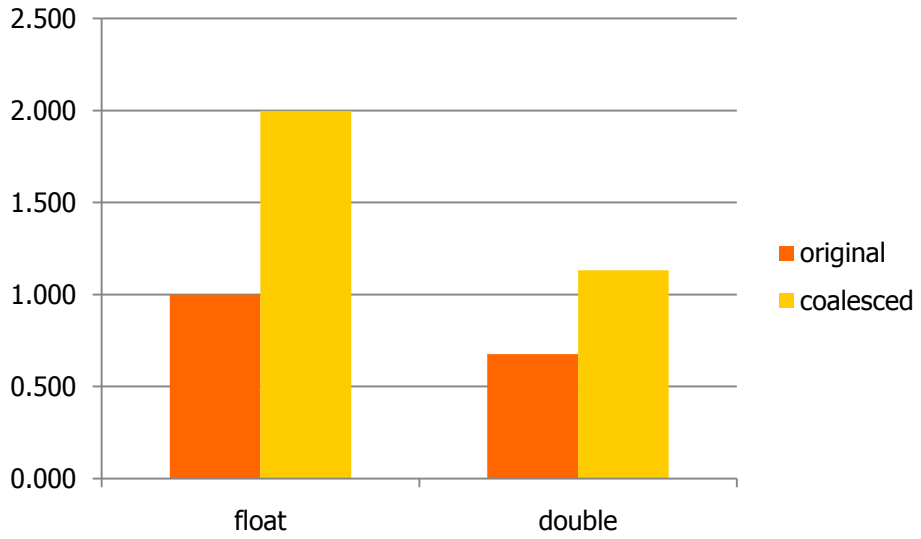
Проблемы реализации

- Каждая нить последовательно читает и пишет столбец 3D массива
  - Коэффициенты и правая часть

- Y, Z – прогонки coalesced
- X – прогонка uncoalesced!

Оптимизация прогонки

- X – прогонка
  - Транспонируем входные массивы и запускаем Y-прогонку

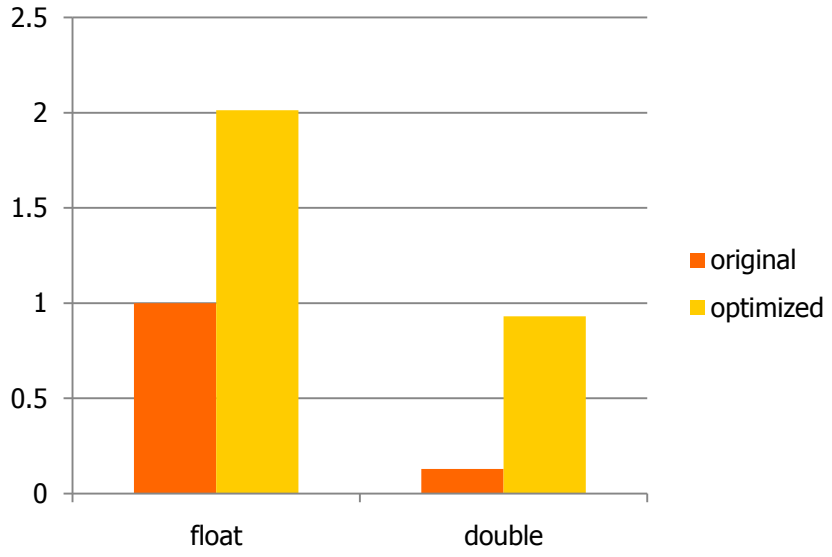


Расчет диссипации

- Расчет частных производных по трем направлениям
  - Локальный доступ к памяти
- Каждая нить обрабатывает столбец данных
  - Переиспользование расчитанных производных
- Использование разделяемой памяти (?)

Оптимизация диссипации

- Рефакторинг кода
  - Предварительный расчет некоторых констант, избавление от лишних if
- C++ шаблоны для X, Y, Z-диссипации
  - Уменьшение числа регистров, нет лишних обращений к памяти



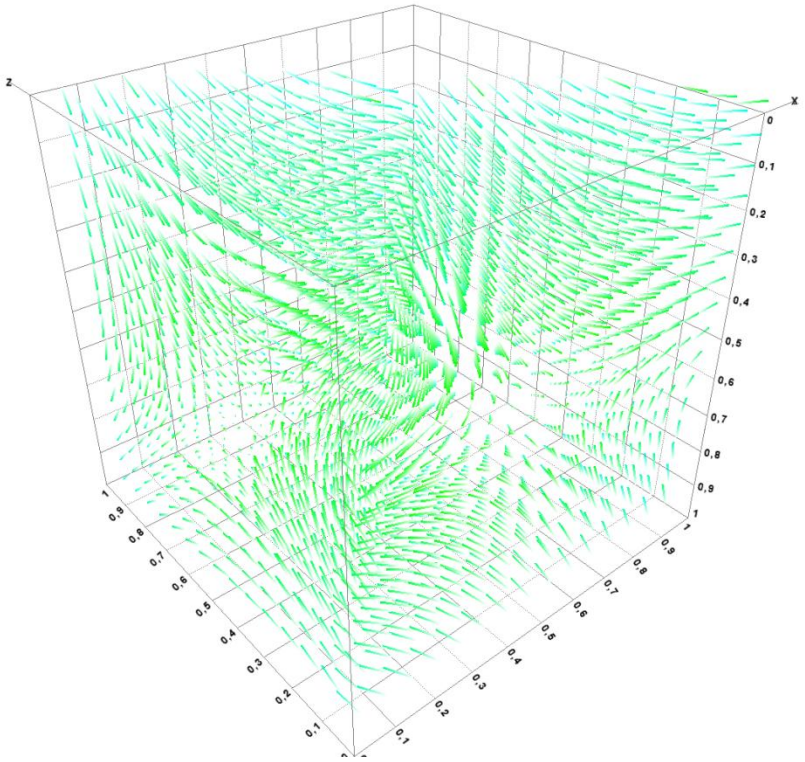
Нелинейные итерации

- Необходимо посчитать полусумму двух 3D массивов
- Каждая нить считает сразу для столбца данных – N^2 нитей
  - Все чтения/записи coalesced
- Оптимальный выбор размера блока
- **80%** от пиковой пропускной способности на Tesla C1060

Реализация на CPU

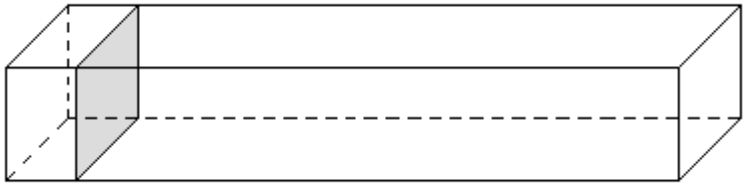
- Использование OpenMP инструкций
- Перестановка циклов для оптимизации работы с кэшем

Результаты

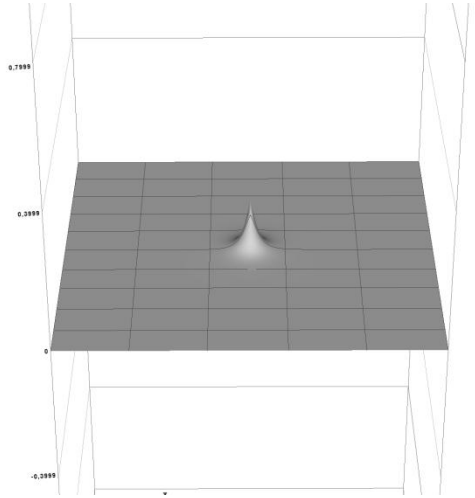


Векторное поле скоростей

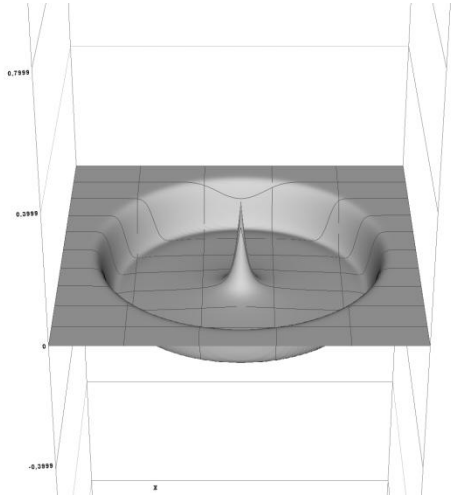
Результаты



Re=250



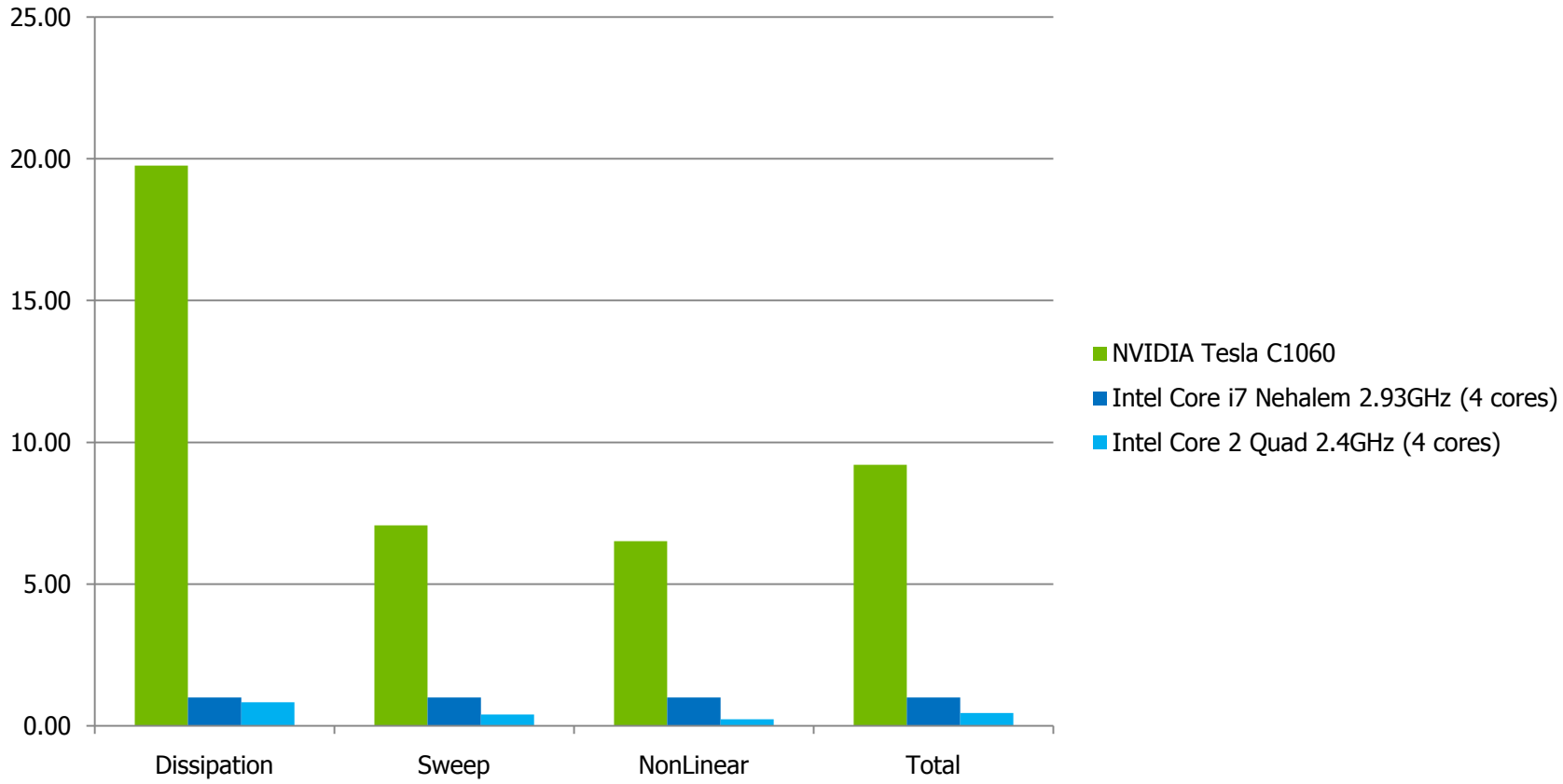
Re=500



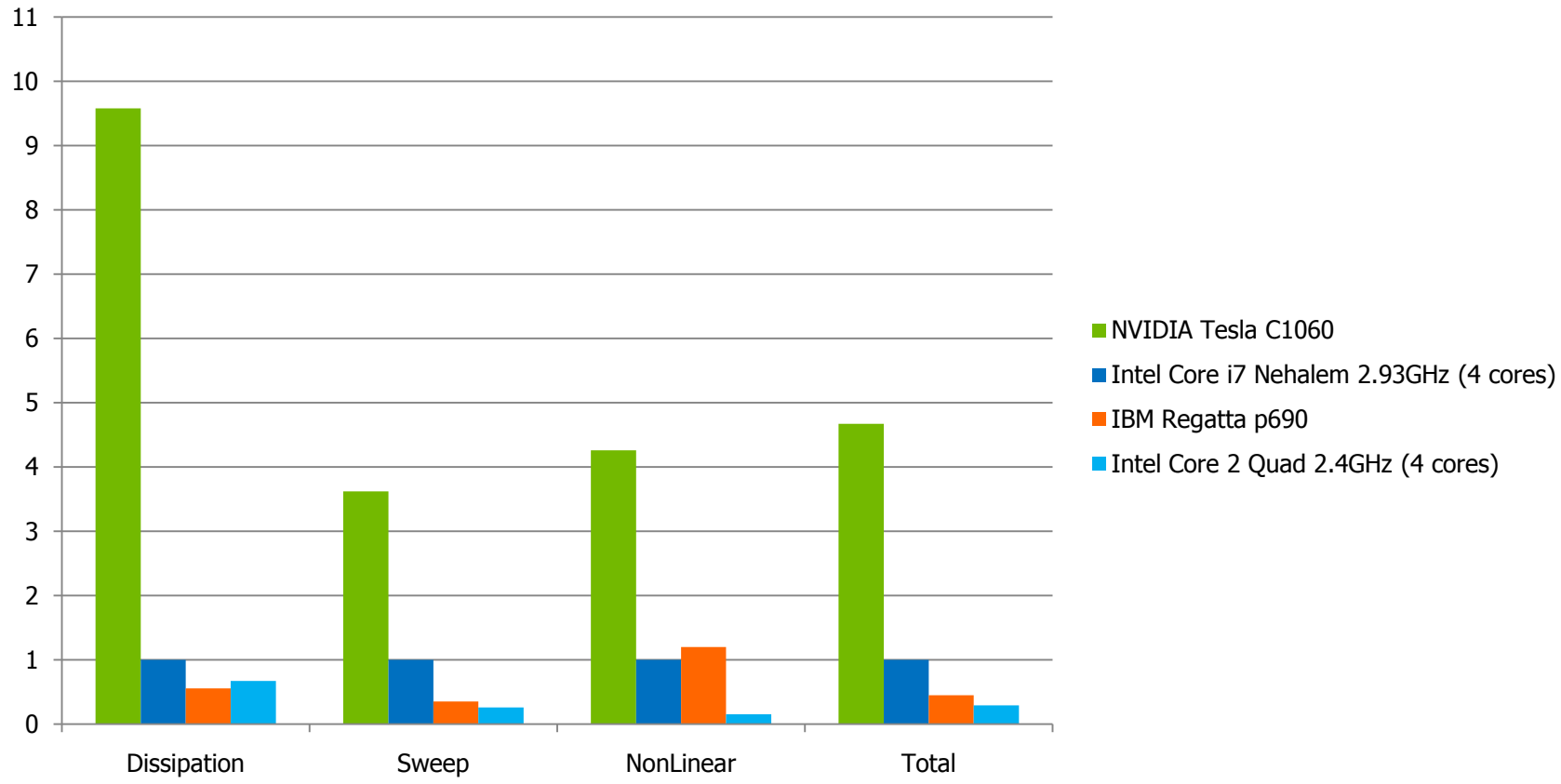
Тест производительности

- Тестовые данные
  - Сетка 128^3
  - 8 нелинейных итераций
- Сравнение CPU, GPU, Regatta
  - Абсолютное время работы
  - Экономическая эффективность

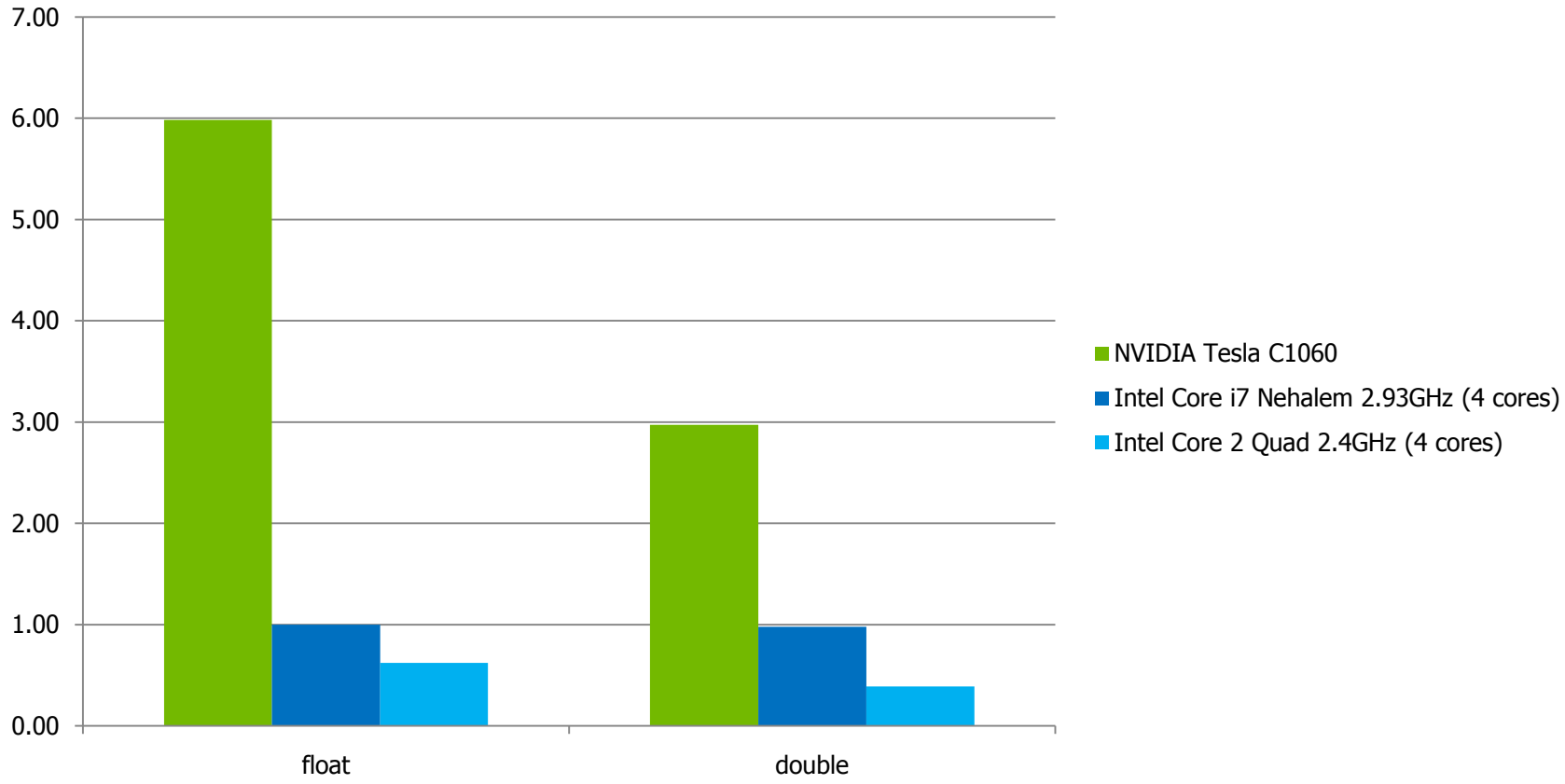
Производительность float



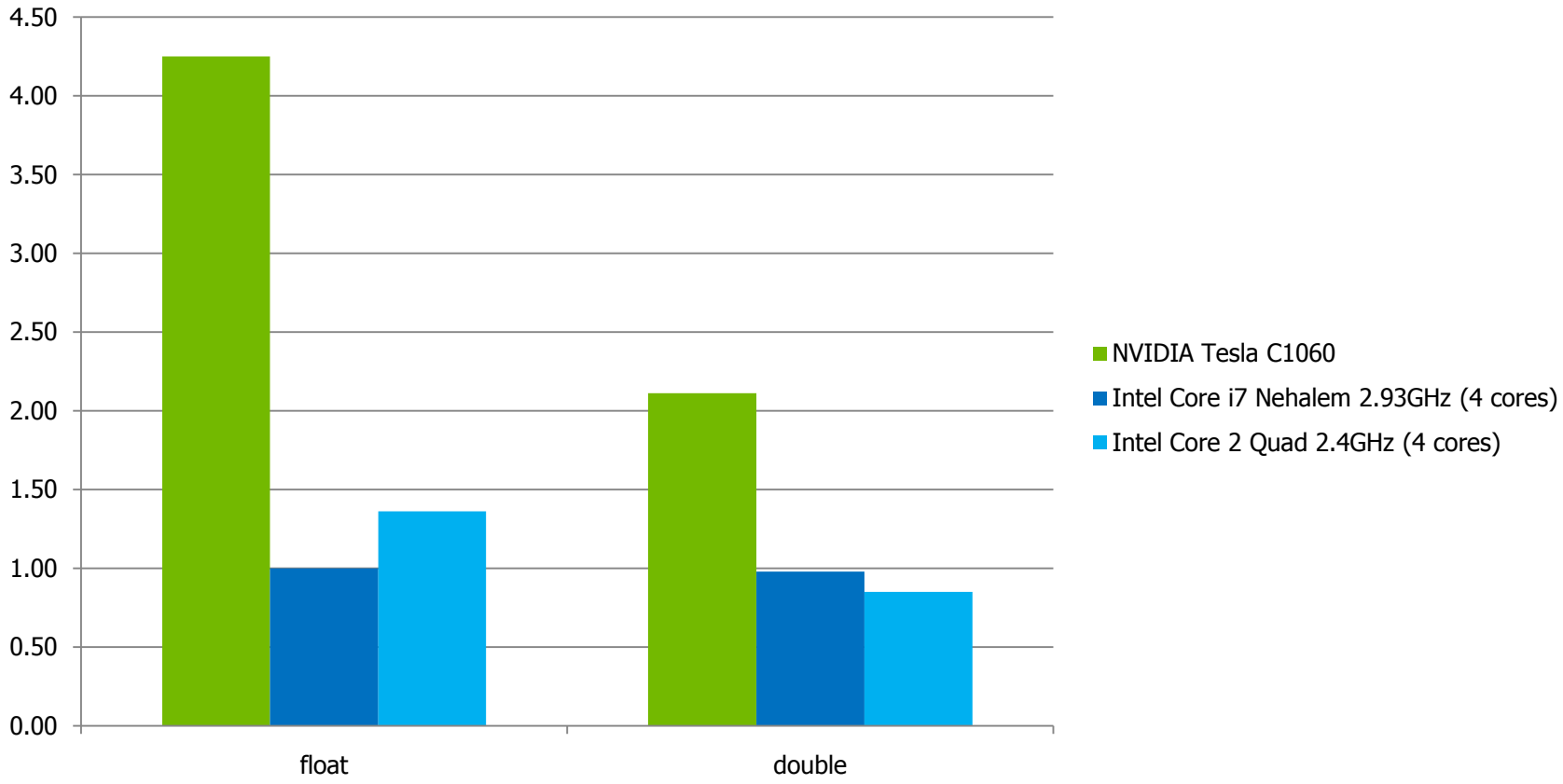
Производительность double



Производительность на W



Производительность на $



Ближайшие планы

- Эффективная реализация на нескольких GPU
  - Расчет на больших сетках с большими числами Рейнольдса
- Оптимизация отдельных ядер
  - Метод редукции для трехдиагональных систем

Выводы

- Высокая эффективность Tesla в задачах аэро-гидродинамики
- Программная модель CUDA – удобное средство утилизации ресурсов GPU
- Применение GPU открывает новые возможности для исследования

Вопросы