



Государственный университет им. Н.И. Лобачевского  
Национальный исследовательский университет

# Макромодульный подход к разработке программ

Докладчик:

Алексей Сиднев

Участники проекта:

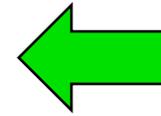
Роман Котельников

Вячеслав Юданов

# Разработка ПО

---

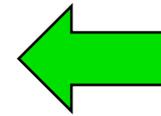
- ❑ Декомпозиция задачи
- ❑ Реализация отдельных модулей



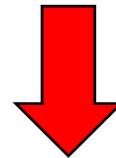
**Модульный подход  
к разработке ПО**

# Разработка ПО

- ❑ Декомпозиция задачи
- ❑ Реализация отдельных модулей



**Модульный подход  
к разработке ПО**



**Недостатки**

- Сильная зависимость от конкретной реализации библиотеки
- Проблема выбора наиболее оптимальной библиотеки под текущие задачи проекта или поддержка нескольких библиотек
- Проблема миграция на новую библиотеку

# Проблемы разработки ПО

---

- Требуется тратить значительные усилия на:
  - переход на новую библиотеку;
  - поддержку нескольких библиотек.



# Переход на новую библиотеку

---

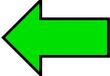
1. Ручная замена используемых библиотек:
  - изменение настроек проекта (lib-библиотеки);
  - изменение заголовочных файлов и путей к ним;
  - изменение прототипов и имён функций;
  - изменение используемых типов данных.



# Переход на новую библиотеку (Примеры)

1. Ручная замена используемых библиотек:

– Матричное умножение (sgemm):

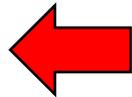
- MKL,  28 строк кода
- cuBLAS.

# Переход на новую библиотеку (Примеры)

## 1. Ручная замена используемых библиотек:

### – Матричное умножение (sgemm):

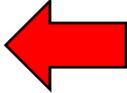
- MKL,
- cuBLAS.



Замена lib-файлов, h-файлов, прототипа и имени функции, дополнительные **14!!!** строк кода, **ЧАС!!!** на переписывание

# Переход на новую библиотеку (Примеры)

## 1. Ручная замена используемых библиотек:

- Матричное умножение (sgemm):
  - MKL,
  - cuBLAS.
- Быстрое преобразование Фурье:
  - FFTW (интерфейс FFTW 3.x, FFTW 2.x),
  - MKL (+ собственный интерфейс). 

**Замена lib-файлов, h-файлов, сборка lib-файлов под требуемый интерфейс**

**или**

**Замена lib-файлов, h-файлов,  
использование собственных интерфейсов  
MKL**



# Переход на новую библиотеку

---

1. Ручная замена используемых библиотек;
2. Использование шаблонов проектирования (декоратор):
  - реализация класса, выполняющего требуемый функционал;
  - конвертирование используемых типов данных;
  - «+» все проблемы пункта 1.



# Поддержка нескольких библиотек

---

- Создание нескольких версий кода:
  - макросы;
  - системы для автоматизации процесса сборки (automake).



# Макромодульный подход

---

- Решить рассмотренные проблемы разработки программ позволяет макромодульный подход к разработке ПО



# Постановка задачи

---

- ❑ Разработать программное средство для поддержки макромодульного подхода:
  - средства статического анализа исходных кодов;
  - надстройка к среде разработке Microsoft Visual Studio.
- ❑ Реализовать поддержку библиотеки MKL и cuBLAS на примере матричного умножения.
- ❑ Подготовить демонстрационный проект.

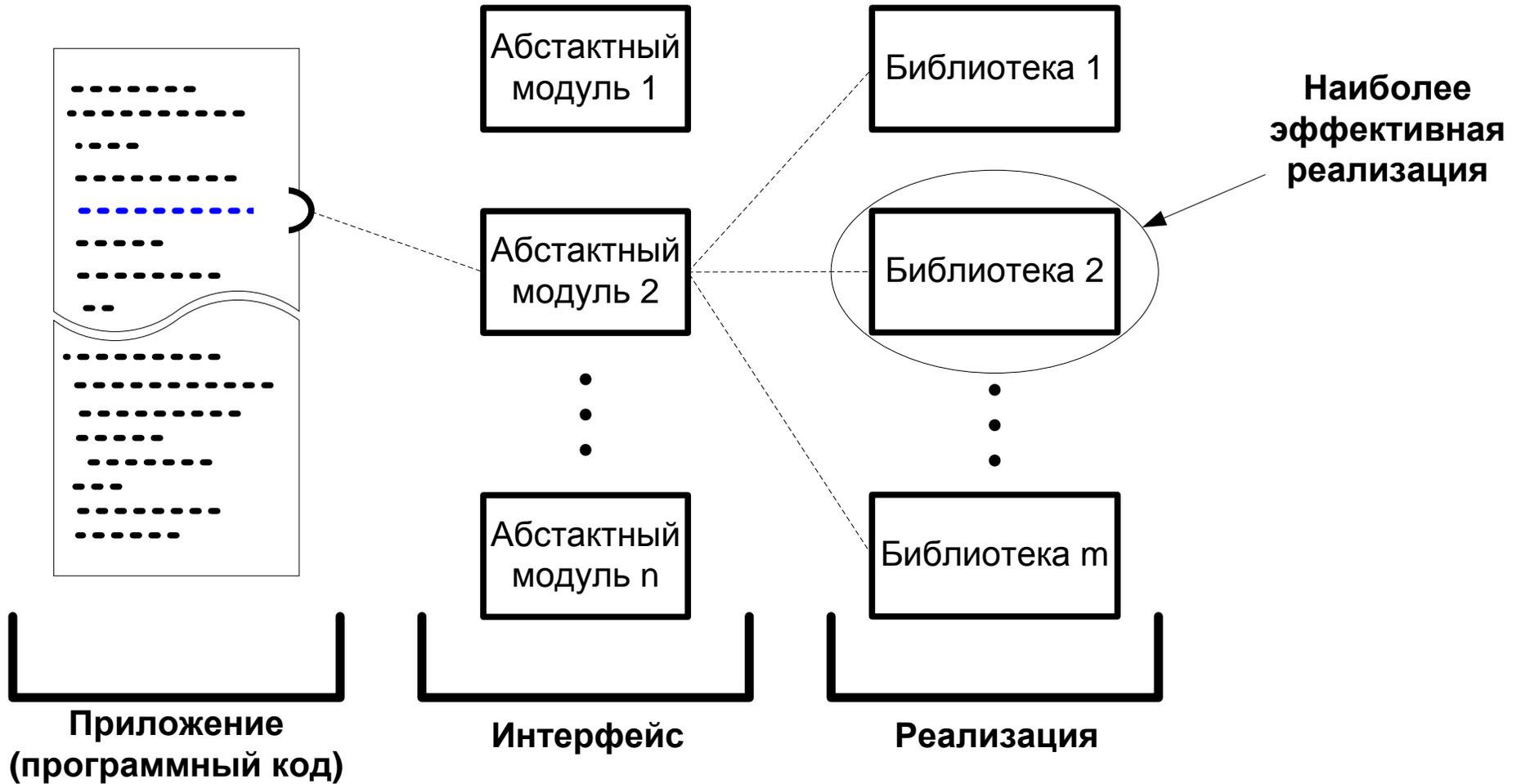


# Общая идея подхода (1)

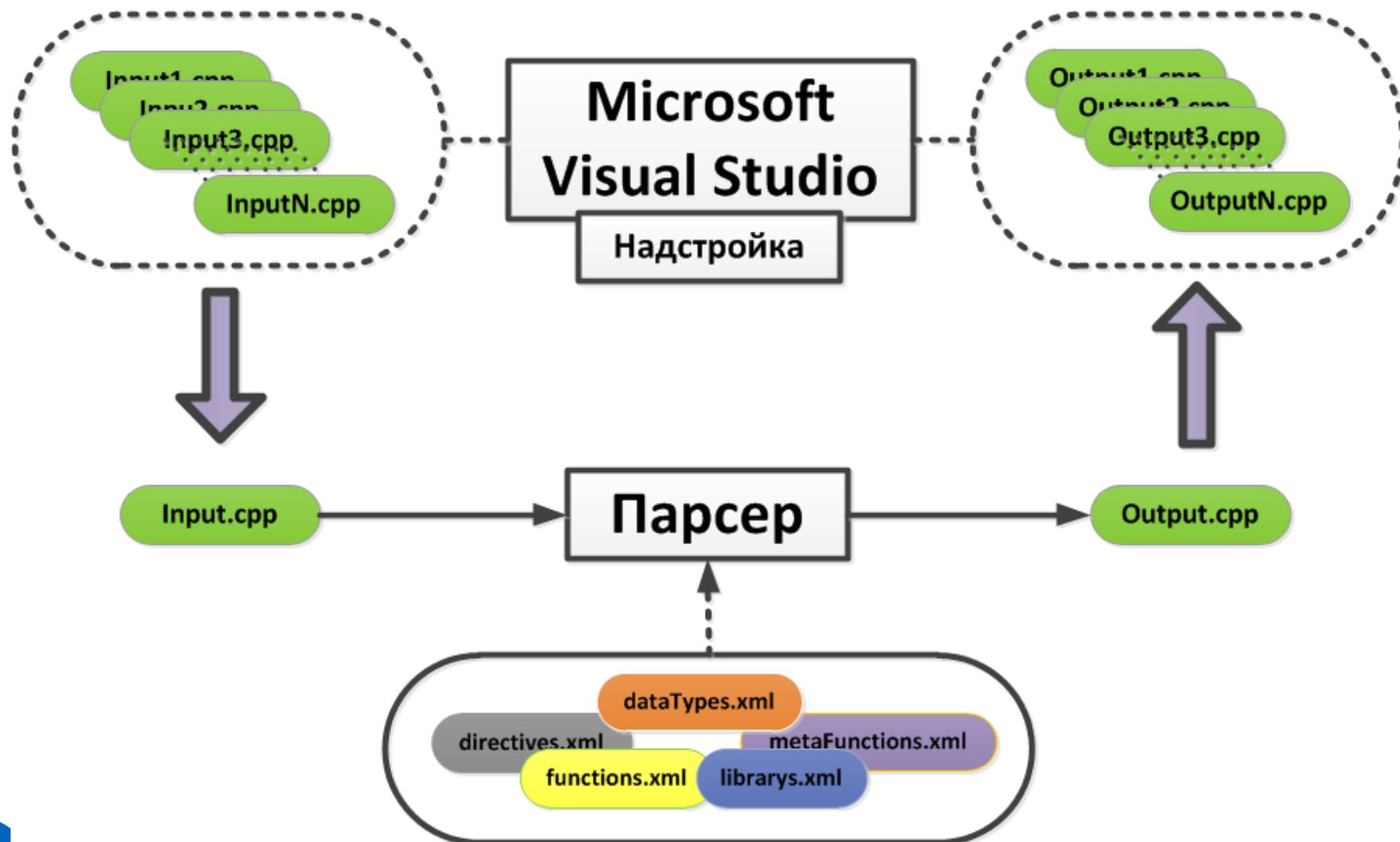
---

- ❑ Разработчик реализует программу с использованием стандартных (абстрактных) модулей
- ❑ Каждый абстрактный модуль имеет реализацию с использованием различных библиотек

# Общая идея подхода (2)



# Архитектура системы



# Компоненты системы

---

- Парсер
  - выполняет статический анализ исходных кодов программы, разбирает описаниями блоков на макроязыке
- Надстройка Microsoft Visual Studio
  - выполняет автоматическую сборку проекта под указанную библиотеку
- Описательные файлы
  - содержат описание поддерживаемых библиотек, реализаций, директив и типов данных
  - имеют формат XML



# Парсер

---

- ❑ Является консольным приложением на языке C#
- ❑ Использует регулярные выражения
- ❑ Необходимая информация для выполнения анализа берётся из описательных файлов
- ❑ Вход:
  - файл с исходными кодами программы
- ❑ Выход:
  - изменённый файл генерируемый в соответствии с выбранной библиотекой



# Настройка Microsoft Visual Studio

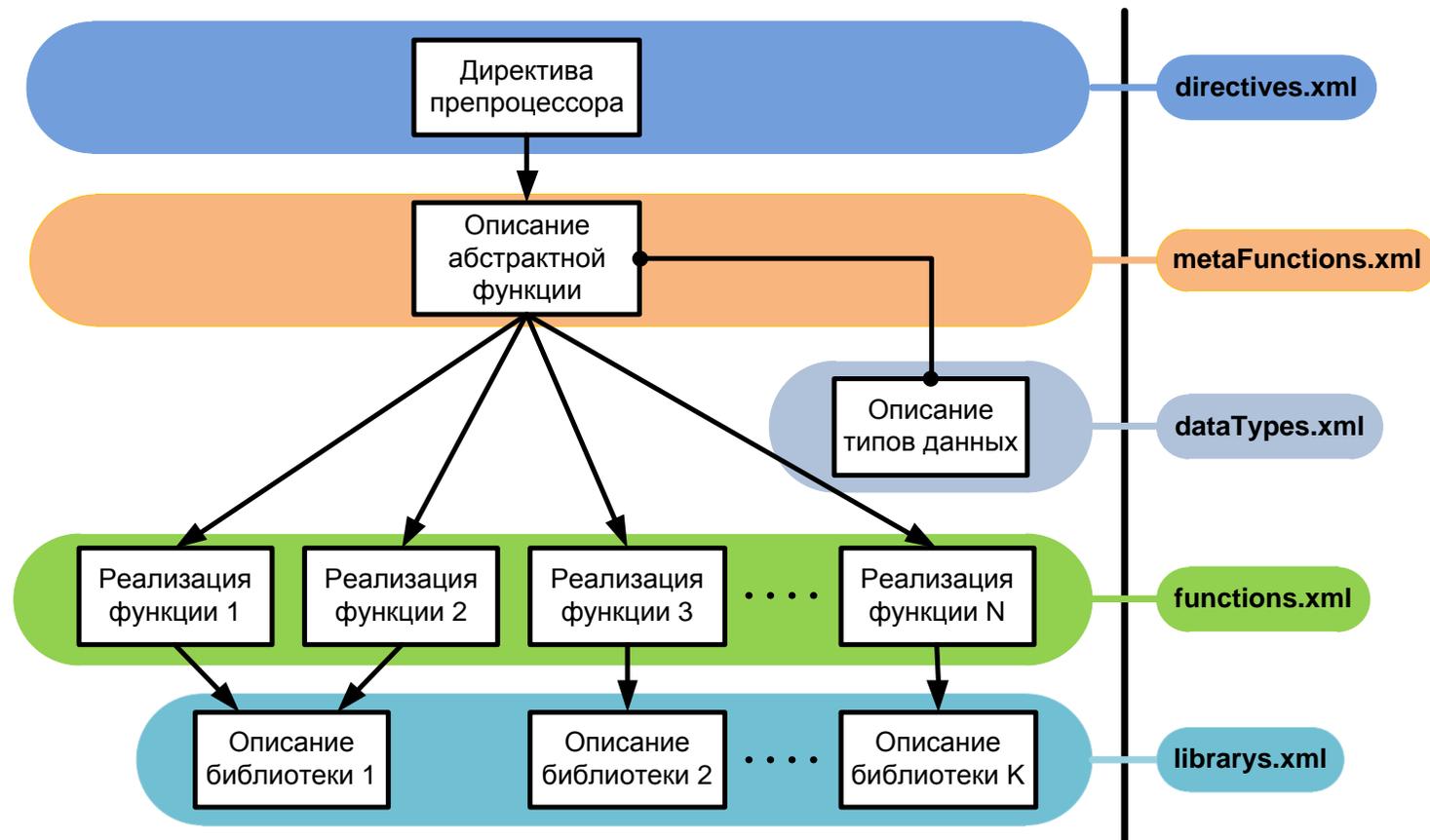
---

- ❑ Позволяет задавать библиотеку для сборки
- ❑ Выполняет автоматическую сборку проекта под указанную библиотеку
- ❑ Для каждого исходного файла в проекте выполняет вызов парсера для генерации новой версии кода



# Описательные файлы

- Содержат описание поддерживаемых библиотек, реализаций, директив и типов данных



# Использование подхода (1)

---

1. Пользователь описывает участки кода программы с помощью директив препроцессора.
2. Пользователь задаёт целевой платформы для сборки.

Автоматически выполняется:

- ❑ Препроцессирование исходных кодов программы.
- ❑ Замена размеченных блоков на реализации из библиотек.



# Использование подхода (2)



# Макроописание (исходная программа)

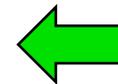
## «Матричное умножение»

```
for(i = 0; i < n; i++)  
  for(l = 0; l < n; l++)  
    for(j = 0; j < n; j++)  
      C[i*n+j] += A[i*n+l] * B[l*n+j];
```



# Макроописание (модификация)

```
#pragma mmt mmult(A=Matrix<CStyle, float>(n, n),  
    B=Matrix<CStyle, float >(n, n), C=Matrix<CStyle, float >(n, n))  
{  
    for(i = 0; i < n; i++)  
        for(l = 0; l < n; l++)  
            for(j = 0; j < n; j++)  
                C[i*n+j] += A[i*n+l] * B[l*n+j];  
}
```



**Не обязательный  
код**

# Тестовые системы

---

## □ Миникластер:

- 2 процессора по 4 ядра (2 x Intel Xeon E5320);
- 4 GB;
- Windows Server 2008 x64;
- интегрированная графика.

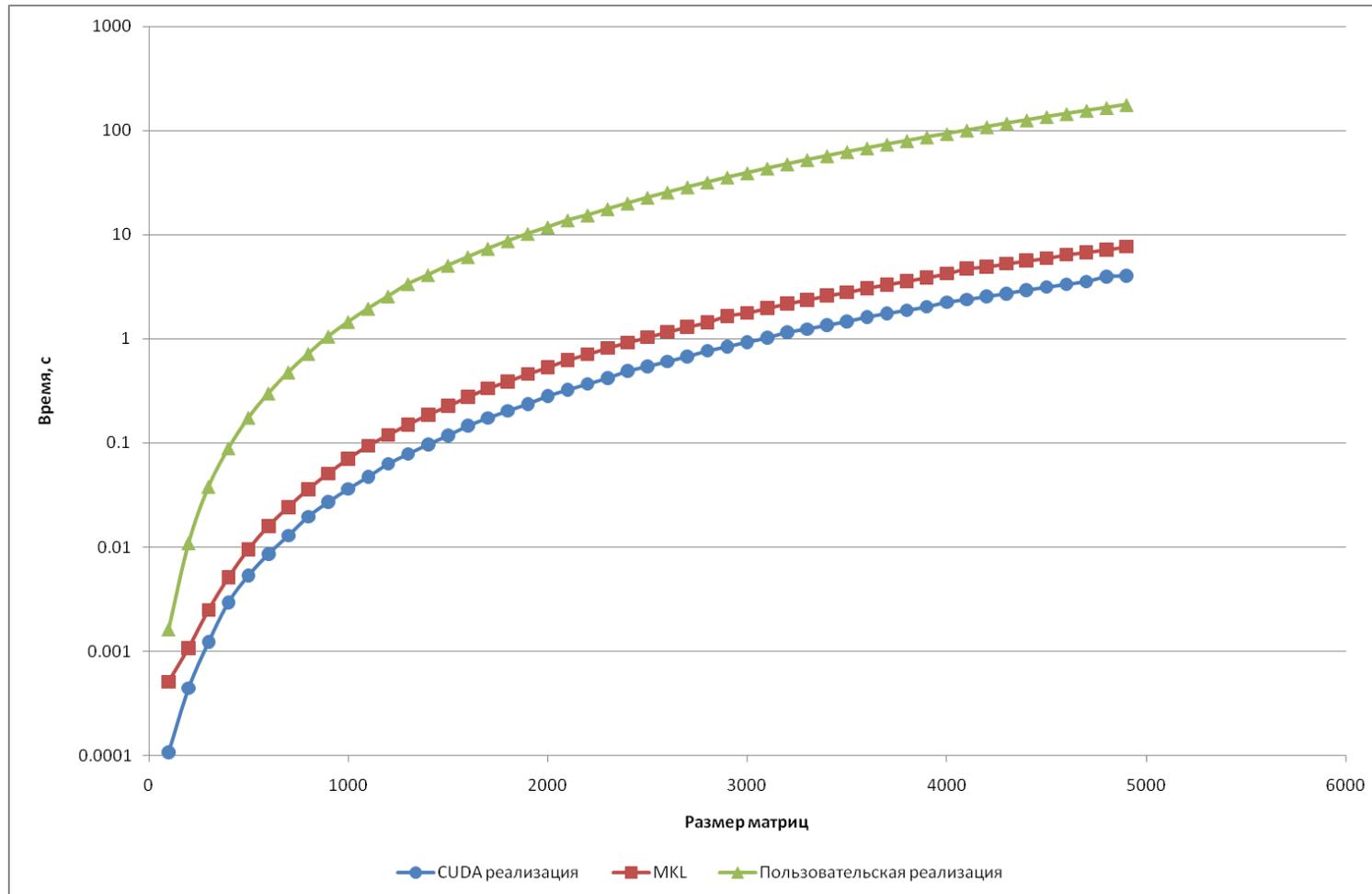
## □ Рабочая станция

- двухъядерный процессор (Intel Core 2 Duo E6550);
- 4 GB;
- Windows 7 x64;
- NVidia GeForce 8800 GTS.



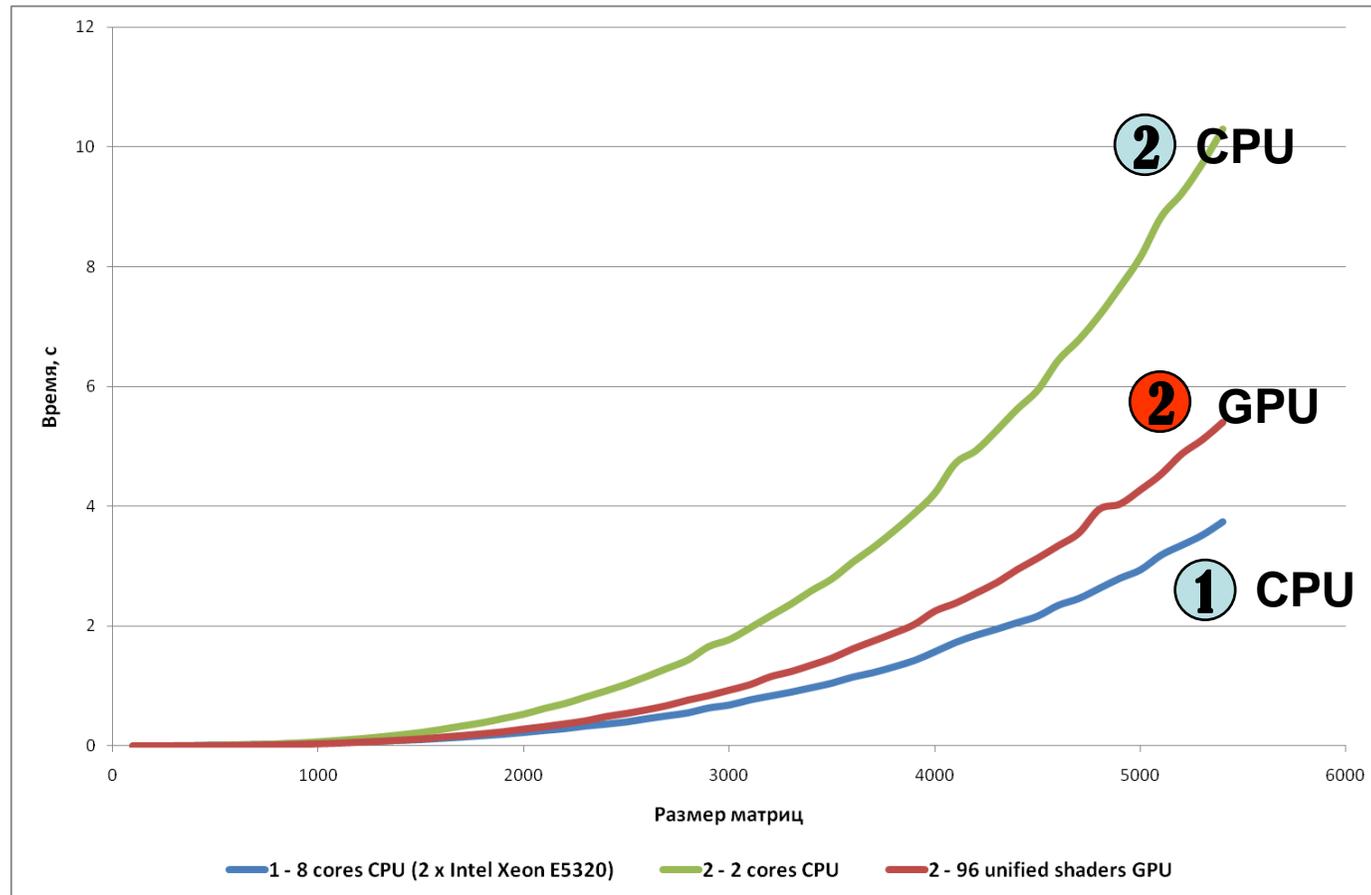
# Результаты экспериментов (1)

- Три версии программы из одного исходного файла.



# Результаты экспериментов (2)

- На разных платформах библиотеки показывают различную эффективность.



# Результаты выполнения проекта

---

- ❑ Разработан прототип системы, реализующий поддержку макромодульной технологии в виде надстройки к Microsoft Visual Studio
- ❑ Разработано средства статического анализа исходных кодов
- ❑ Реализована поддержка библиотеки MKL и cuBLAS
- ❑ Разработано демонстрационное приложение



# Публикации

---

- В.П. Гергель, А.А. Сиднев. О применении макромодульной технологии разработки программ // Высокопроизводительные вычисления на кластерных системах. Материалы X научной конференции. Том 1. Пермь, 2010. С. 172-178.

# Дальнейшие планы

---

- ❑ Разработка стандарта описательных файлов
- ❑ Реализация поддержки новых библиотек и функций
- ❑ Автоматический выбор реализации библиотеки:  
статический и во время исполнения

# Вопросы

---

□ ???

