ДВФУ
ДАЛЬНЕВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Исследование алгоритмов БПФ и их параллельных форм

Выполнил: студент группы М9123-09.04.04рпис, А.В. Марков

Научный руководитель: Доцент департамента ПИиИИ, к.т.н., А.А. Чусов

Владивосток 2025

Актуальность работы

- БПФ — **основной инструмент** в цифровой обработке сигналов: изображения, аудио, видео, радиосвязь
- Объем обрабатываемых данных стремительно растет — требуется всё более быстрые алгоритмы для анализа
- Классические алгоритмы БПФ плохо масштабируются — важно адаптировать их **под многопроцессорные архитектуры**

Цель и задачи

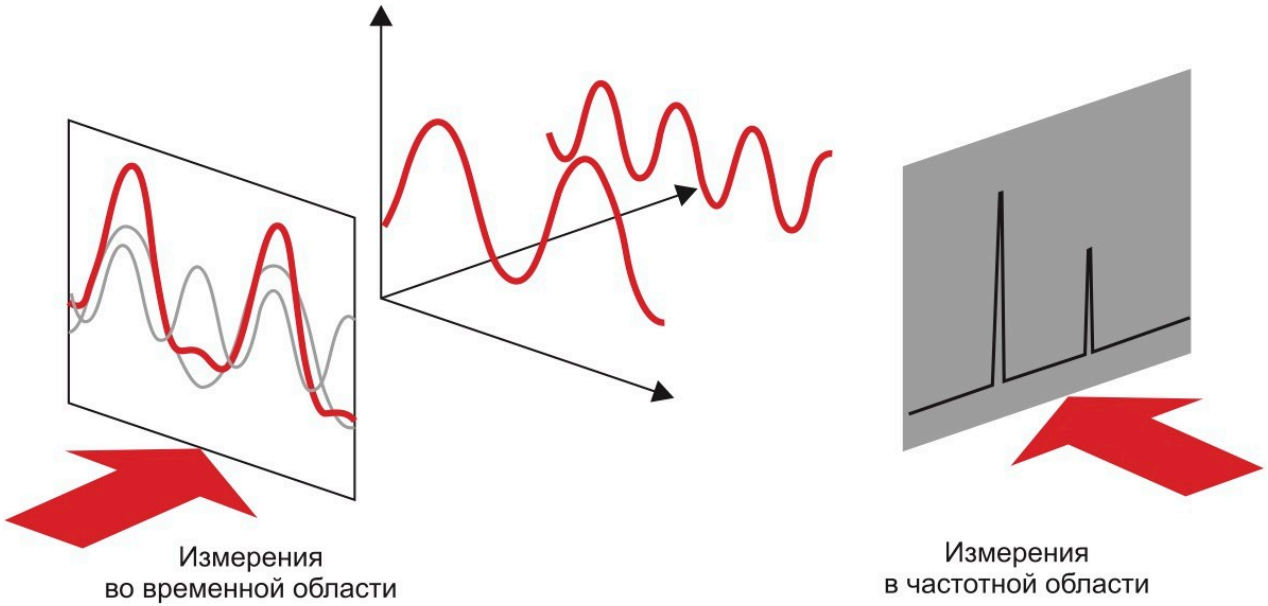
Цель:

Разработка и исследование параллельных реализаций алгоритмов БПФ.

Задачи:

1. Подготовить обзор алгоритмов БПФ и существующих библиотек
2. Выполнить анализ предметной области
3. Разработать математические модели параллельных схем БПФ
4. Реализовать алгоритмы и выполнить эксперименты
5. Исследовать производительность в различных ситуациях
6. Сформулиовать практические выводы по результатам исследования

Принцип работы ДПФ



Прямое ДПФ: $X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-2\pi i \frac{kn}{N}}, \quad k = 0, 1, \ldots, N-1$

Обратное ДПФ: $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{2\pi i \frac{kn}{N}}, \quad n = 0, 1, \ldots, N-1$

Обзор существующих алгоритмов

Алгоритм	Ключевые особенности
Кули–Тьюки	Базовый алгоритм; эффективен при длине $N = 2^k$
Райдера	Предназначен для простых чисел N; преобразует ДПФ в свёртку
Блюстайна	Работает с произвольной длиной N
Стокхама	Итеративная структура; эффективен по памяти и легко распараллеливается
Гуда–Томаса	Основан на КТО; не требует поворотных множителей; эффективен при взаимно простых множителях $N = N_1 \cdot N_2$

Сравнение библиотек БПФ

Характеристика \ Библиотека	FFTW	cuFFT	TurboFFT	NumPy/SciPy	PyCUDA/Numba
Поддержка CPU	+	–	–	+	–
Поддержка GPU	–	+	+	–	+
Высокая производительность	+	+	+	–	+
Оптимизация под архитектуру	+	–	+	–	–
Параллелизм (многопоточность)	+	–	+	+ (BLAS)	–
Аппаратная оптимизация	–	+	+	–	+

Метрики оценки эффективности

1. Время выполнения:

$$T_{total} = O\left(\frac{C}{P}\right) + T_{comm}(N, P)$$

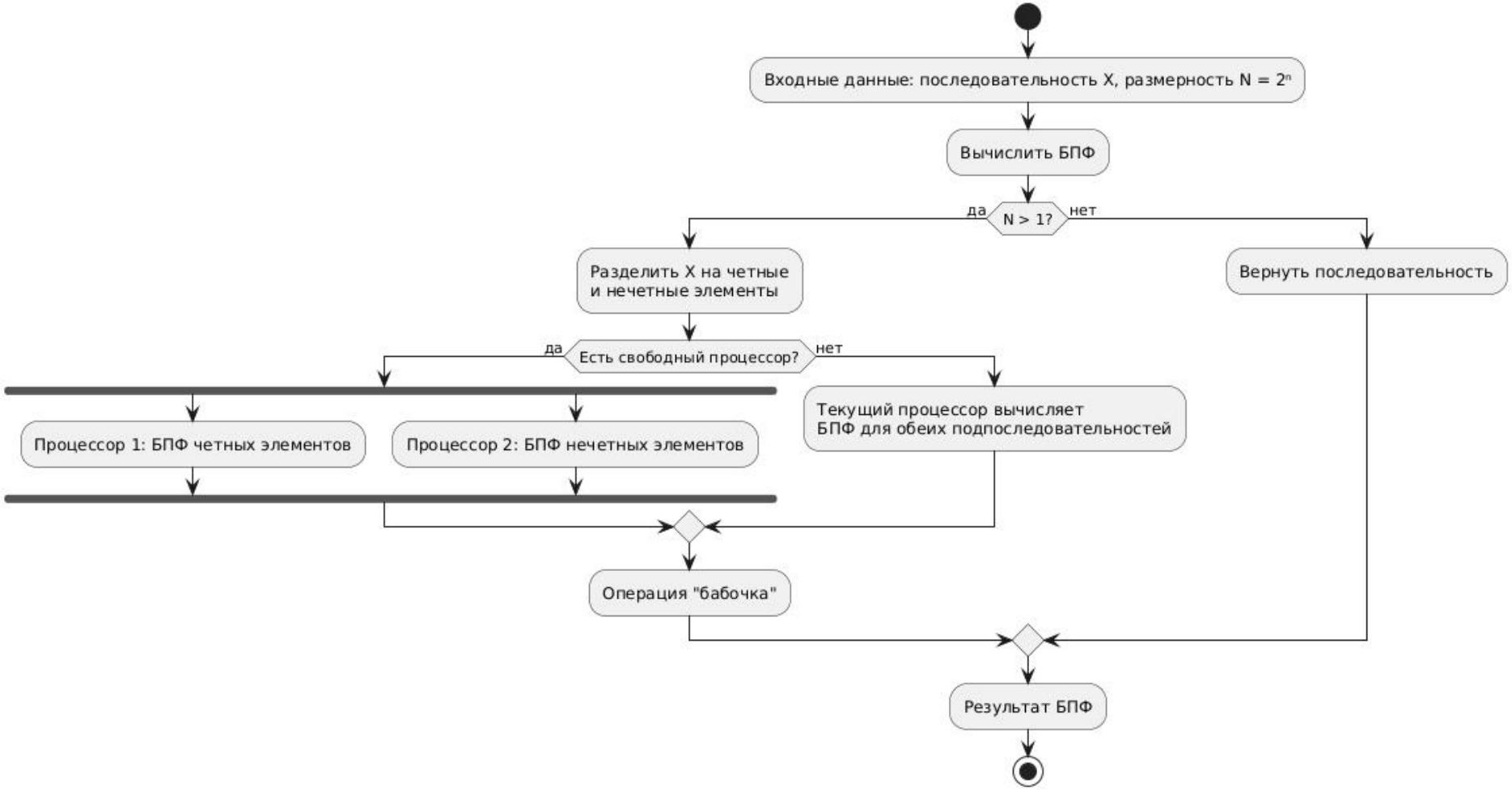
2. Энергопотребление CPU:

$$E_{total}(P) = T_{total} \cdot P \cdot E_{CPU}$$

3. Энергоэффективность CPU:

$$\eta(P) = \frac{C}{E_{total}(P)}$$

Разработанные схемы



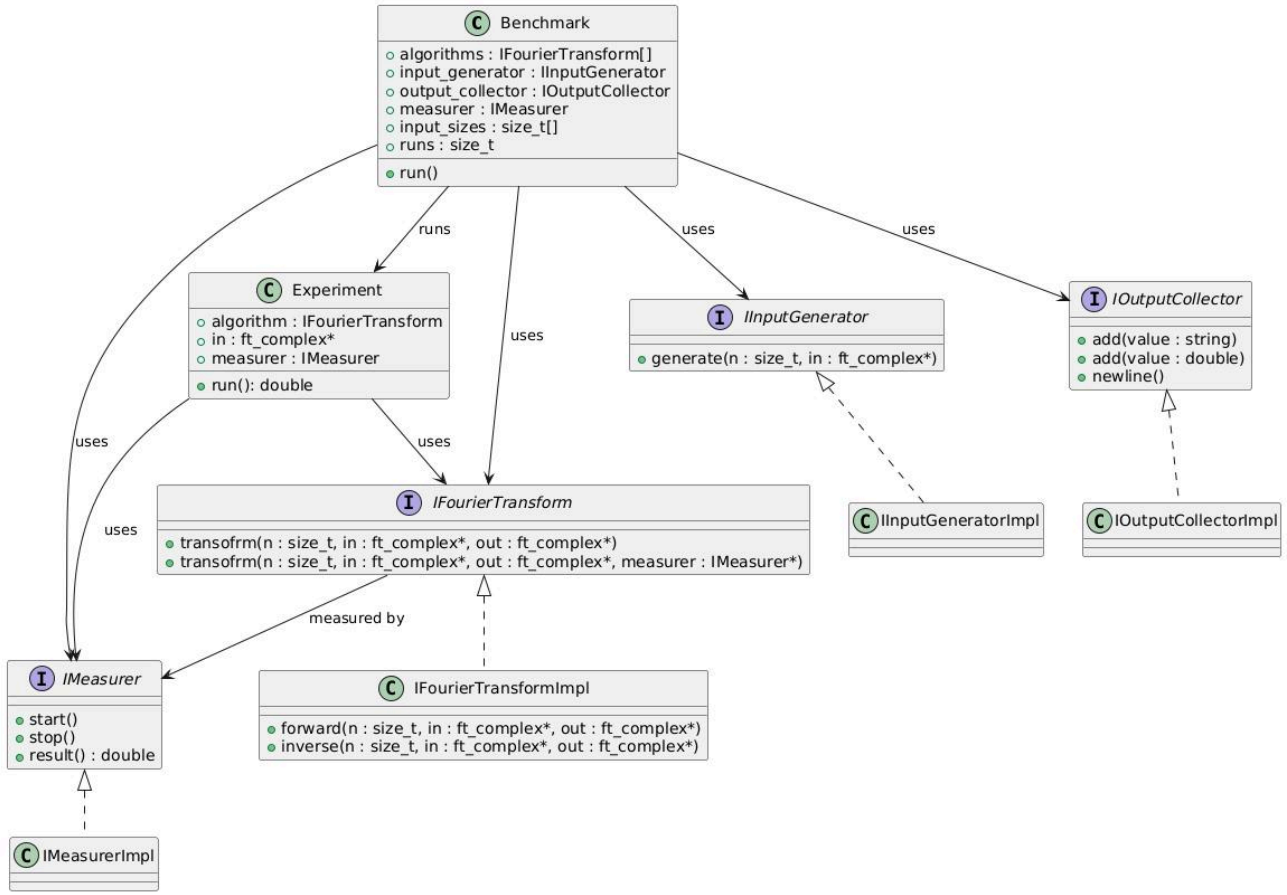
Архитектура системы

- Язык реализации: C++
- Поддержка многопоточности (OpenMP + `std::thread`)
- Визуализация и анализ результатов в Jupyter Notebook
- Компоненты:
  - Генератор последовательностей
  - Исполнитель алгоритмов
  - Измерители (время, энергопотребление)

Реализация

- Объектно-ориентированный подход (C++)
- Интерфейс алгоритмов: `IFourierTransform`
- Интерфейс измерителей: `IMeasurer`
- Визуализация при помощи библиотек: `matplotlib`, `pandas`

Архитектурно-контекстная диаграмма



Экспериментальное тестирование

- Наборы данных:
  - Синусоидальные сигналы
  - Случайные и импульсные последовательности
- Метрики:
  - Время выполнения
  - Энергопотребление

Исследование

- Масштабируемость с увеличением числа потоков
- Эффективность алгоритмов:
  - При малых N — Стокхам
  - При больших N — Кули–Тьюки с произвольным основанием
- Энергосбережение с приближенными методами

Заключение

1. Подготовлен обзор алгоритмов БПФ и существующих библиотек
2. Выполнен анализ предметной области
3. Разработаны математические модели параллельных схем БПФ
4. Реализованы алгоритмы и выполнить эксперименты
5. Исследована производительность в различных ситуациях
6. Сформулированы практические выводы по результатам исследования

Цель достигнута, задачи решены

Практическая и научная значимость

- Практическая:
  - Оптимизация БПФ для многопроцессорных систем
  - Возможность адаптации под разные архитектуры
- Научная:
  - Расширение моделей оценки эффективности
  - Анализ приближённых методов и их применимости

Спасибо за внимание!