

Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра системного программирования

# Автоматизация разработки параллельных программ с помощью библиотеки легковесных нитей пользовательского уровня

Рябыкин Владислав Александрович, 427 группа

**Научный руководитель:**

Катаев Никита Андреевич

Москва, 2025

# Введение

## **Argobots: A Lightweight Low-level Threading Framework**

Многопоточная библиотека на С для разработки параллельных программ

Плюсы:

1. Открытый исходный код (OpenSource);
2. Гибкость настройки;

Минусы:

1. Отсутствие операции редукции;
2. Отсутствие планировщика с кражей задач.

# Постановка задачи

**Цель работы** – добавить новую функциональность в библиотеку Argobots, позволяющую использовать её для автоматизации разработки параллельных программ.

## **Задачи:**

1. Разработать и программно реализовать алгоритм для редукции в Argobots;
2. Разработать и программно реализовать алгоритм планировщика с кражей задач в Argobots;
3. Написать тесты для проверки полученных реализаций;
4. Проанализировать полученные результаты запуска тестов.

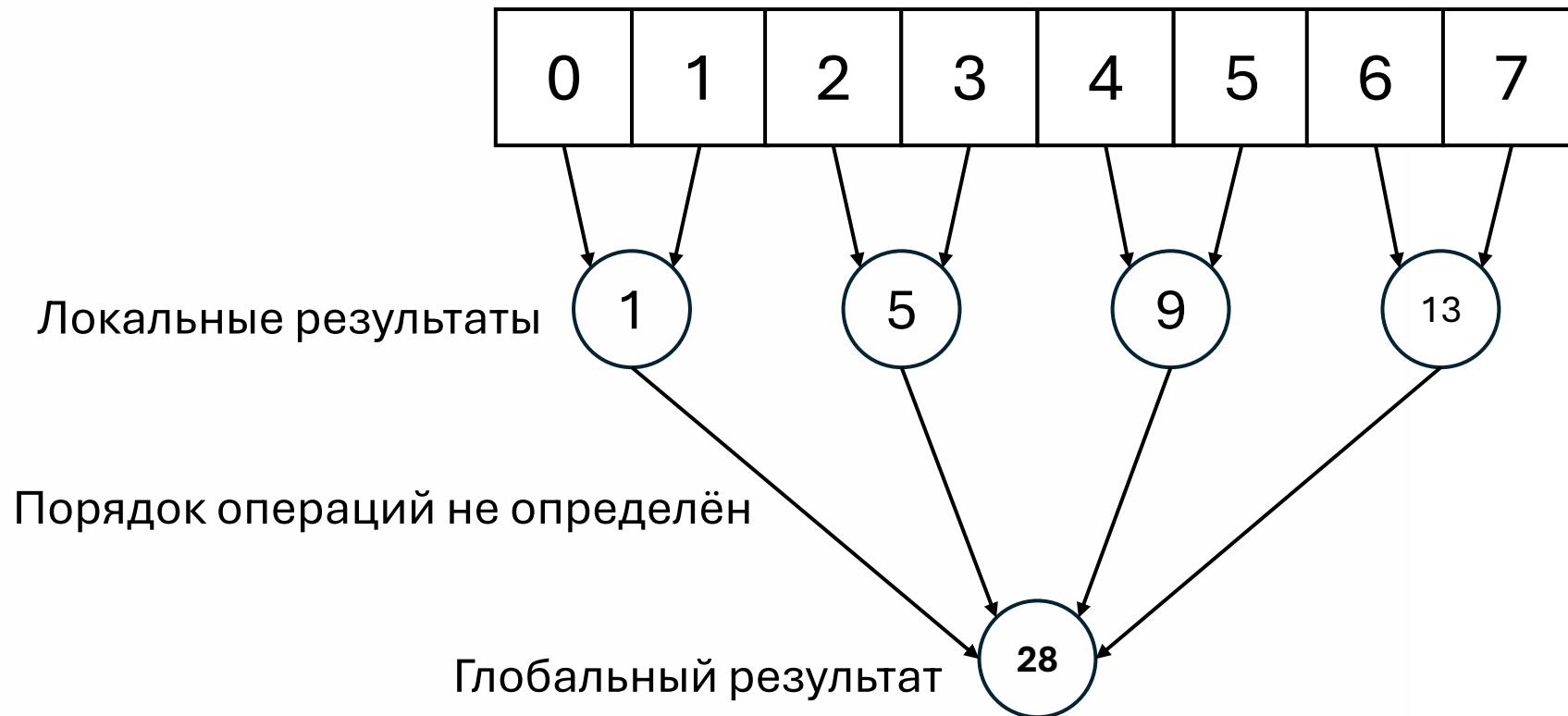
# Редукция. Наивная реализация

Редукция своей части массива

**Заблокировать мьютекс**

Редукция глобального и локального результатов

**Разблокировать мьютекс**

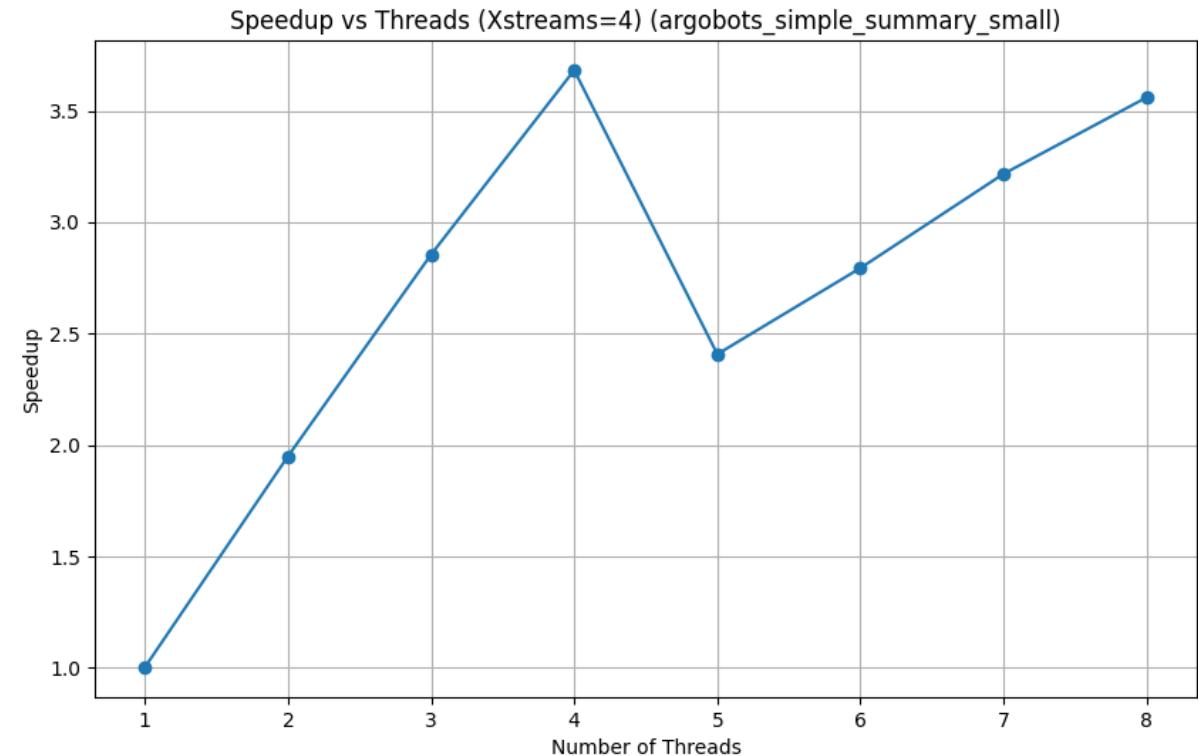
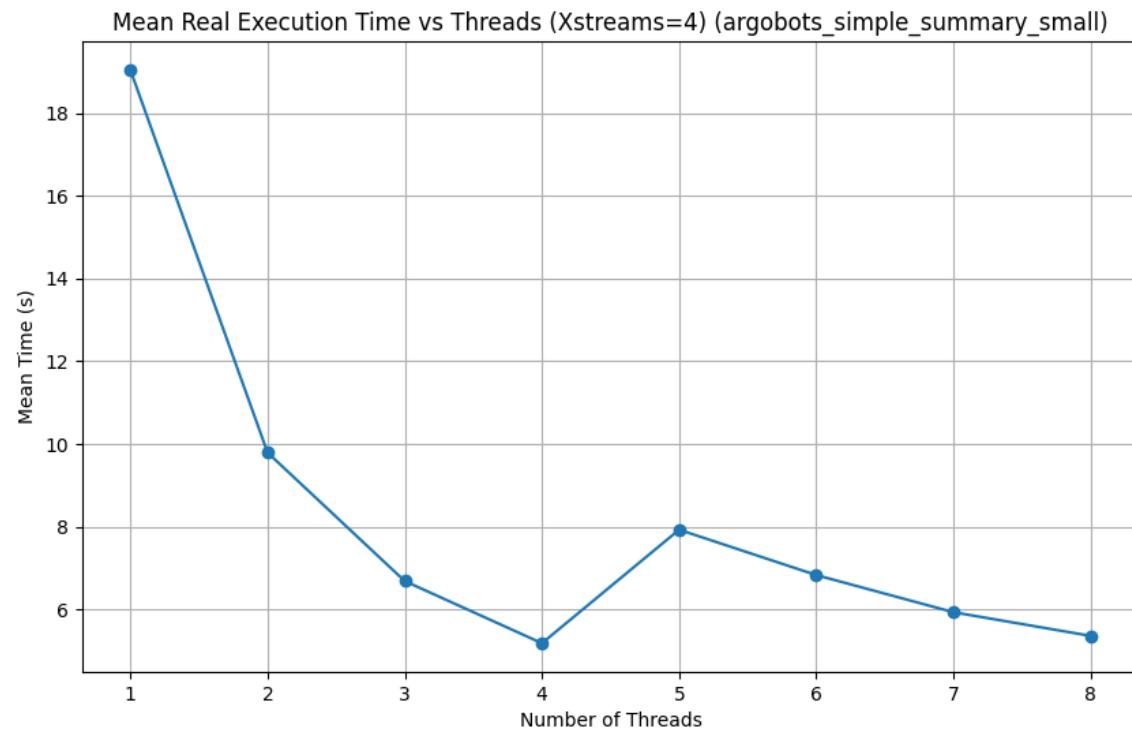


# Редукция. Наивная реализация. Результаты тестов

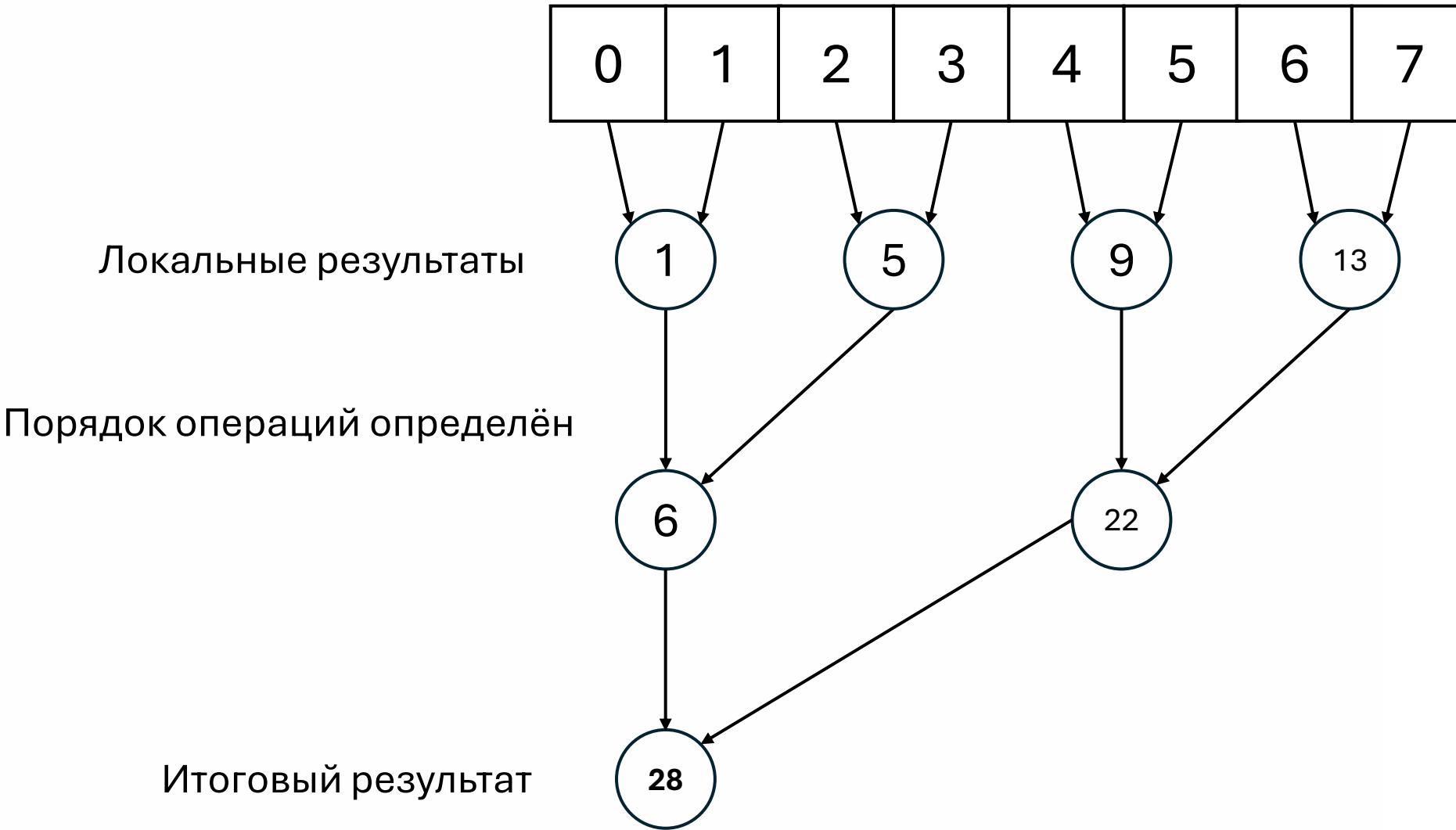
Тесты **jac3d**

Тесты **NAS CG**

Ubuntu 24.04 LTS  
4 ядра Intel 2 ГГц  
16 Гигабайт RAM



# Редукция. Метод сдавивания

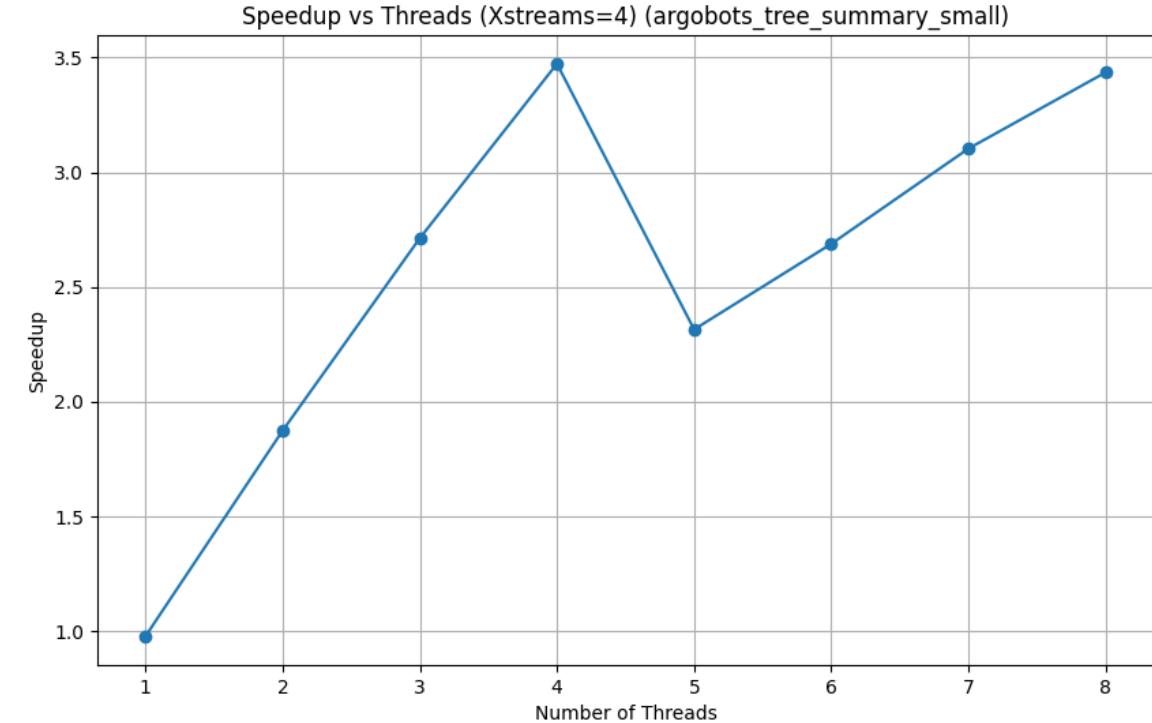
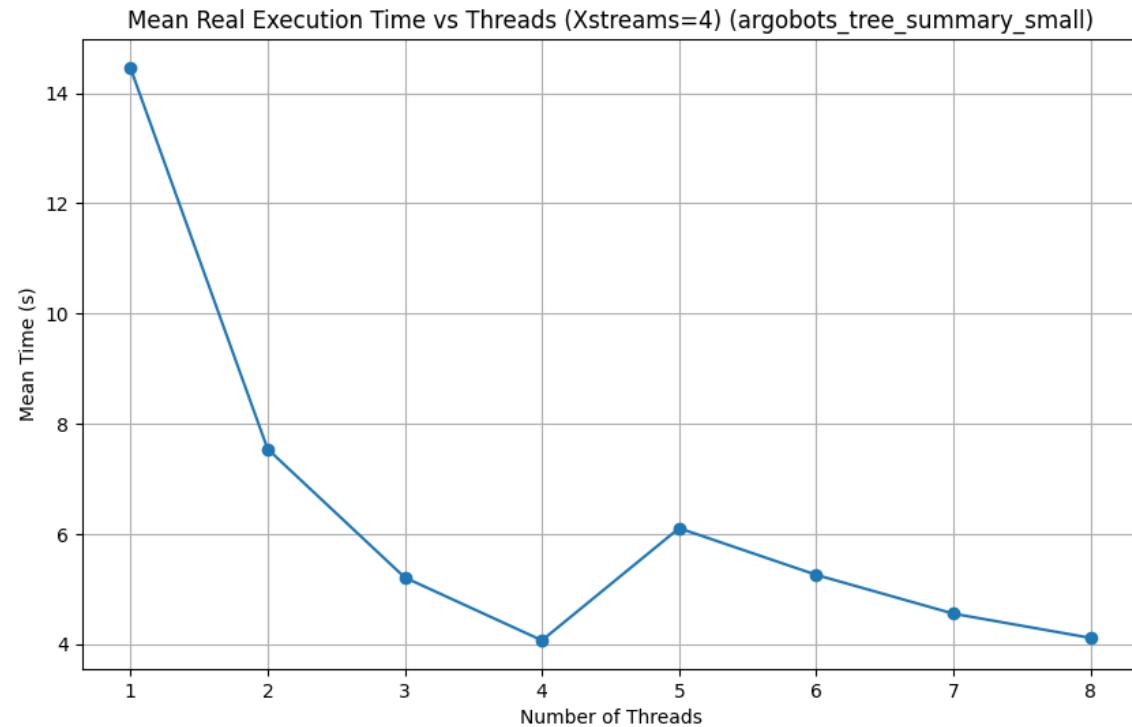


# Редукция. Метод сдавивания. Результаты тестов

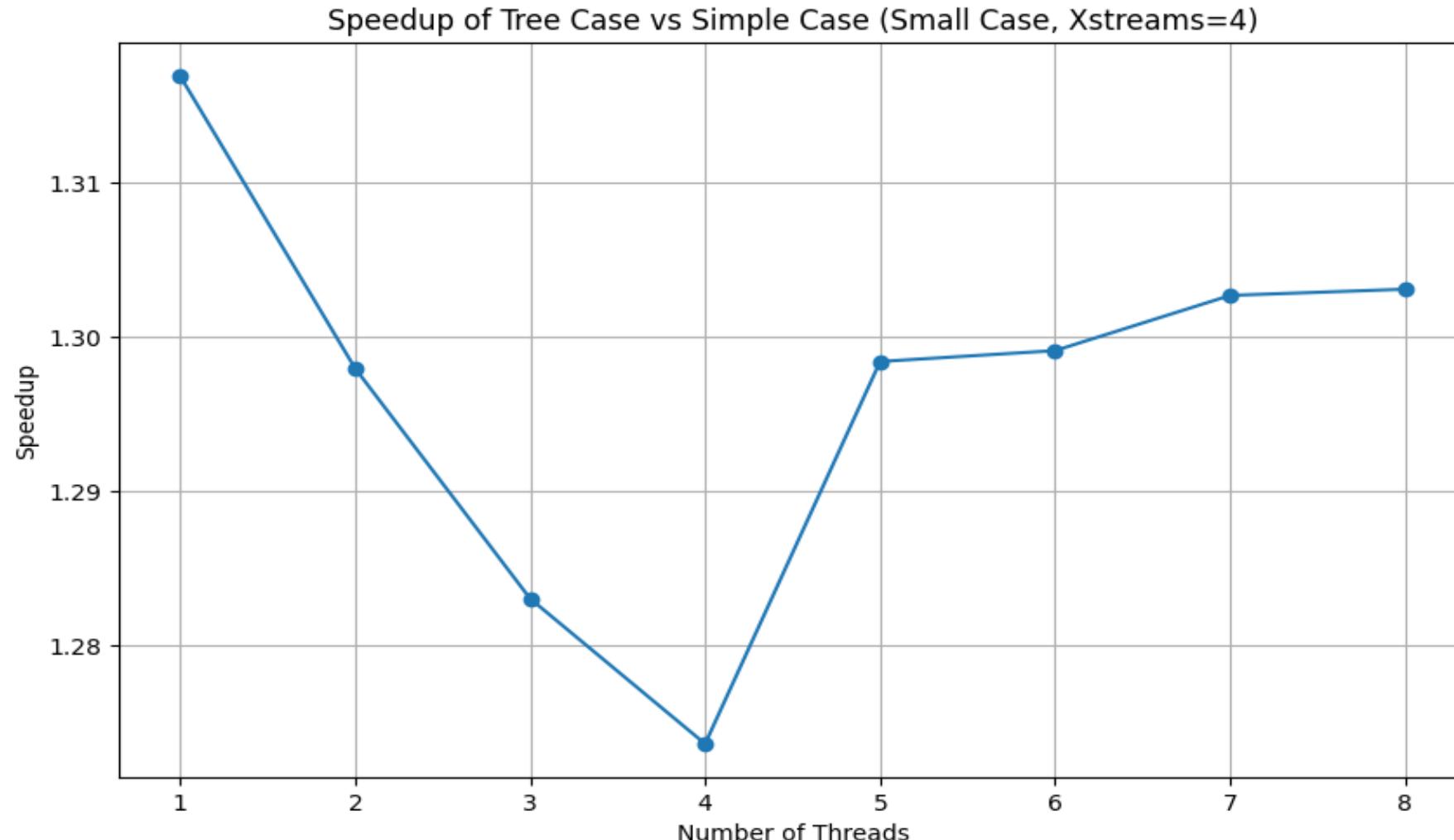
Тесты **jac3d**

Тесты **NAS CG**

Ubuntu 24.04 LTS  
4 ядра Intel 2 ГГц  
16 Гигабайт RAM



# Редукция. Сравнение результатов методов



# Редукция. Результаты

1. С использованием Argobots разработаны и программно реализованы два алгоритма редукции: наивный и методом сдваивания;
2. Тесты **jac3d** и **NAS CG** написаны с использованием Argobots;
3. Метод сдваивания эффективнее наивной реализации на примерно **30%**;
4. Оба метода демонстрируют хорошее распараллеливание программ:  
ускорение **~3.5** раз на 4 ядрах.

# Планировщик

1. Поток берёт задачи для выполнения из пула;
2. У каждого пула есть свой планировщик (scheduler);
3. Стандартная стратегия – брать задачи ТОЛЬКО из своего пула; возможны простой потока;
4. Хотим реализовать кражу работы в случае, когда поток простояивает (work stealing scheduler).

# Планировщик. Схема работы. 1



Пул задач



Ожидающая  
выполнения задача



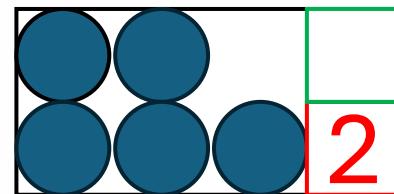
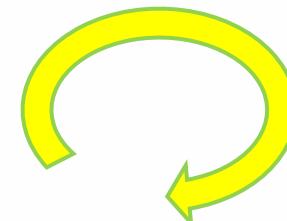
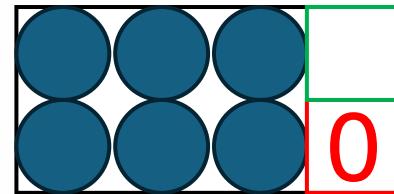
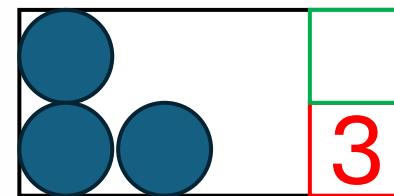
Украденная задача



Номер пула



Направление обхода



# Планировщик. Схема работы. 2



Пул задач



Ожидающая  
выполнения задача



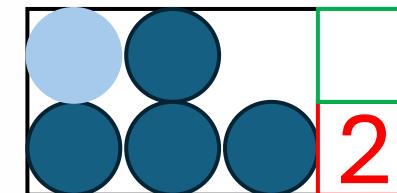
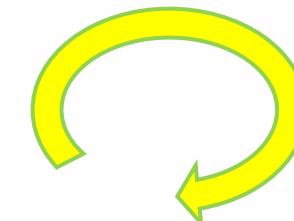
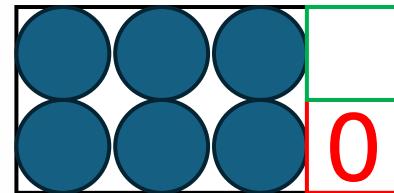
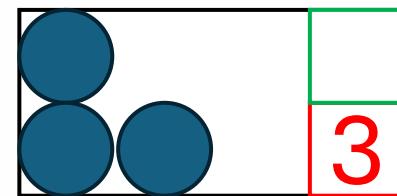
Украденная задача



Номер пула



Направление обхода



Дай задачу

# Планировщик. Схема работы. 3



Пул задач



Ожидающая  
выполнения задача



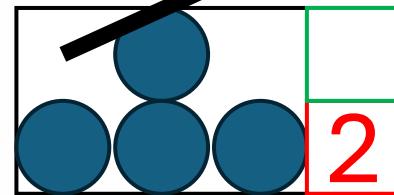
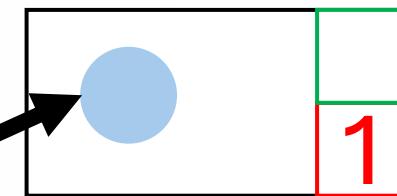
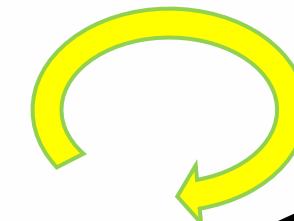
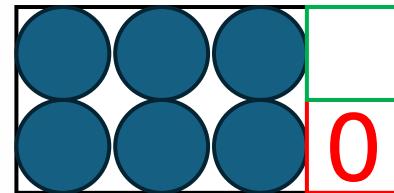
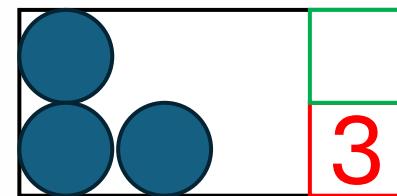
Украденная задача



Номер пула



Направление обхода



# Планировщик. Результаты тестов

1. Здесь появятся результаты запуска тестов **NPB-MZ**;
2. Здесь появится сравнение различных конфигураций:
  1. Последовательная;
  2. OpenMP;
  3. Argobots, стандартный планировщик;
  4. Argobots, написанный планировщик с кражей задач.

# Планировщик. Результаты

1. С использованием Argobots разработан и программно реализован алгоритм планировщика с кражей задач (work stealing scheduler);
2. Здесь будут результаты по итогам запуска тестов **NPB-MZ**.

# Результаты

1. Реализована операция редукции (наивный метод и метод сдваивания) в библиотеке Argobots; метод сдваивания быстрее на ~30%;
2. Реализован планировщик с кражей задач в Argobots;
3. Написаны тесты **jac3d**, **NAS CG** с использованием библиотеки Argobots;
4. Проанализированы результаты запуска тестов. На 4 ядрах достигается ускорение ~3.5 раза.

**Полученная реализация редукции и планировщика позволяет использовать библиотеку Argobots для автоматизации разработки параллельных программ.**

# Дальнейшие планы (слайд для весеннего отчёта)

1. Реализовать тест **NPB-MZ** с использованием написанного в Argobots планировщика;
2. Сравнить результаты работы теста **NPB-MZ**:
  1. Последовательная реализация;
  2. OpenMP;
  3. Argobots, стандартный планировщик;
  4. Argobots, самописный планировщик с кражей задач;
3. Написать текст ВКР;
4. Обновить презентацию.