

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ
УНИВЕРСИТЕТ «МИФИ»»**

Институт Интеллектуальных Кибернетических Систем
Кафедра №42 "Криптология и кибербезопасность"

Дисциплина «Параллельное программирование»

Отчет к лабораторной работе № 1.1
«Введение в параллельные вычисления. Технология OpenMP»

Выполнил:
студент группы Б22-505
Титов Дмитрий Иванович

Принял:
Куприяшин Михаил Андреевич

Москва
2024 год

Цель

Изучить как влияет увеличения числа вычислений на ускорение, эффективность и время работы программы.

Ход выполнения нагрузки

В ходе работы, в параллельной программе дополнительно будет осуществляться нагрузка методом добавления 1000 операций постфиксной инкрементации переменной, операция будет выполняться каждый 1000 элемент, 10 000 элемент, 100 000 и 1 000 000.

1. Полученные данные

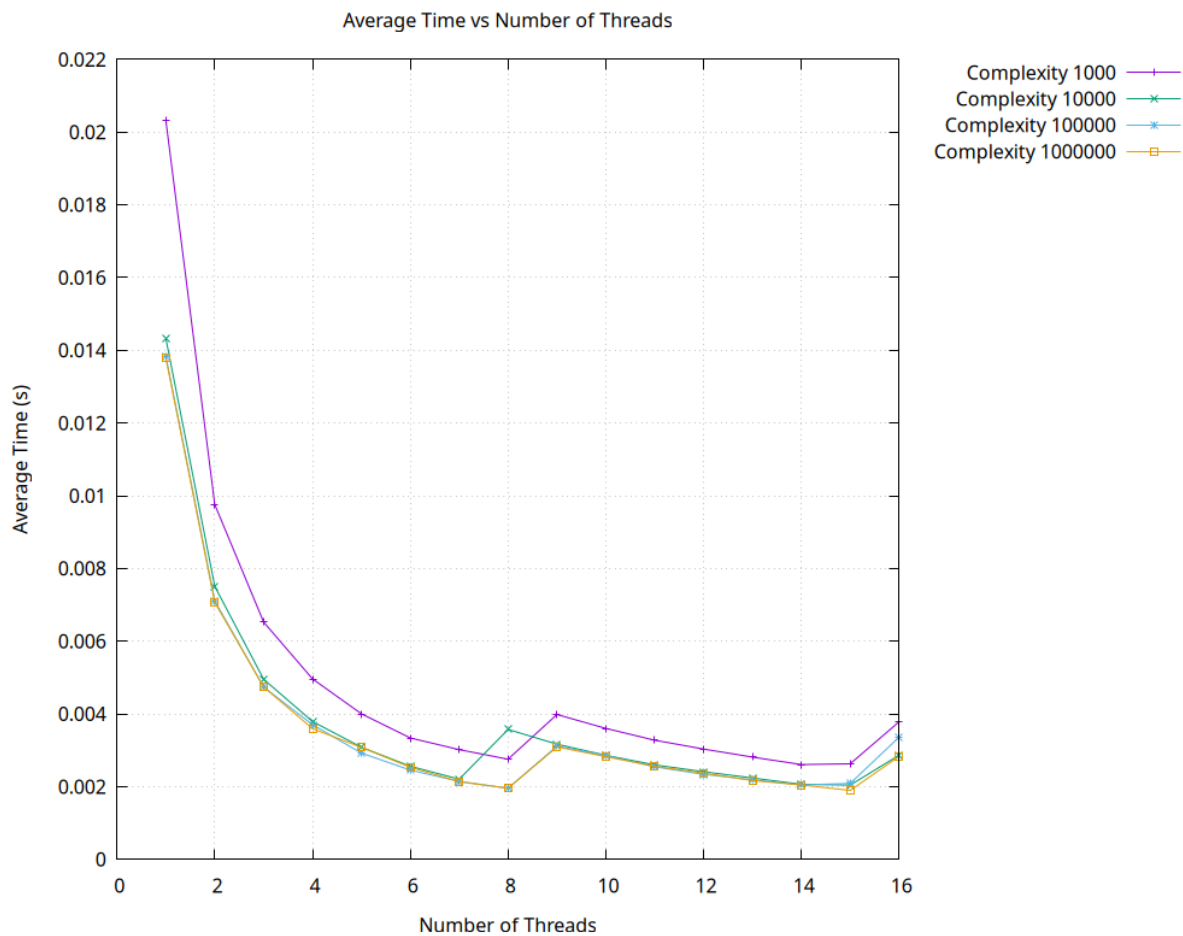


График 1. Среднее время от числа потоков и количества дополнительной нагрузки

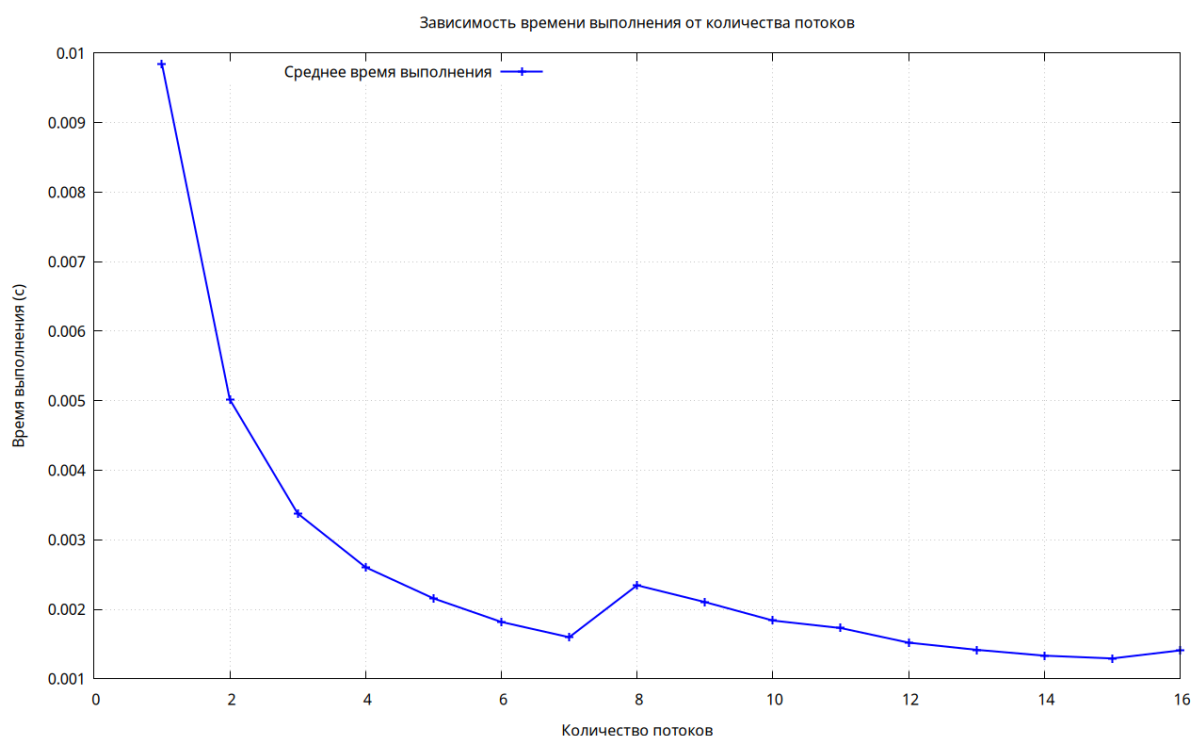


График 2. Среднее время от числа потоков

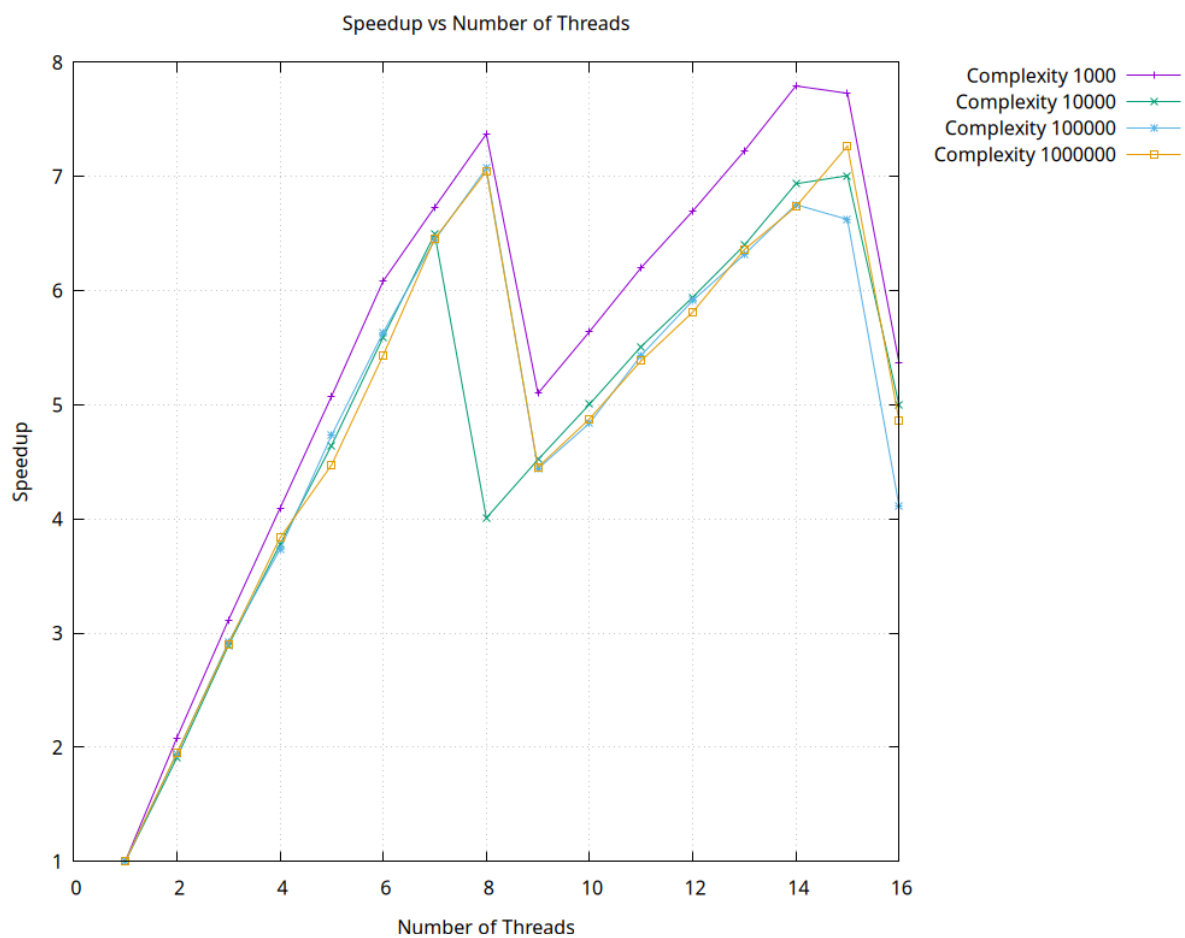


График 3. Ускорение от числа потоков и дополнительной нагрузки

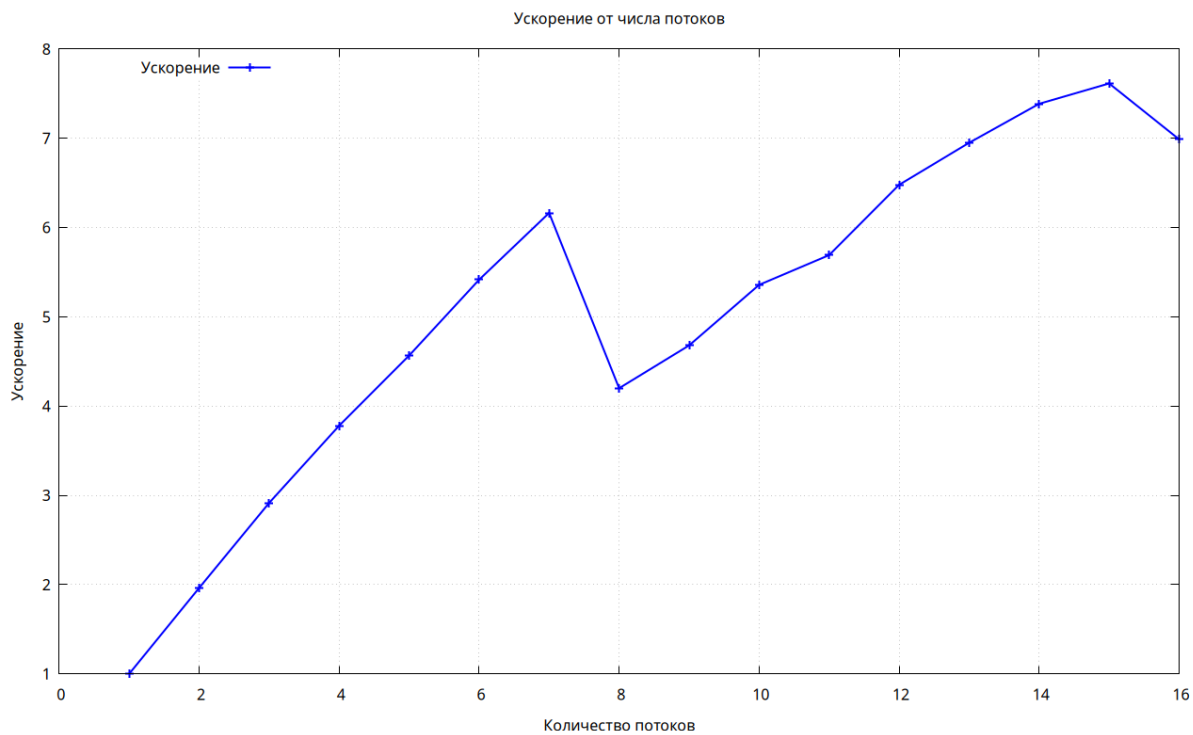


График 4. Ускорение от числа потоков

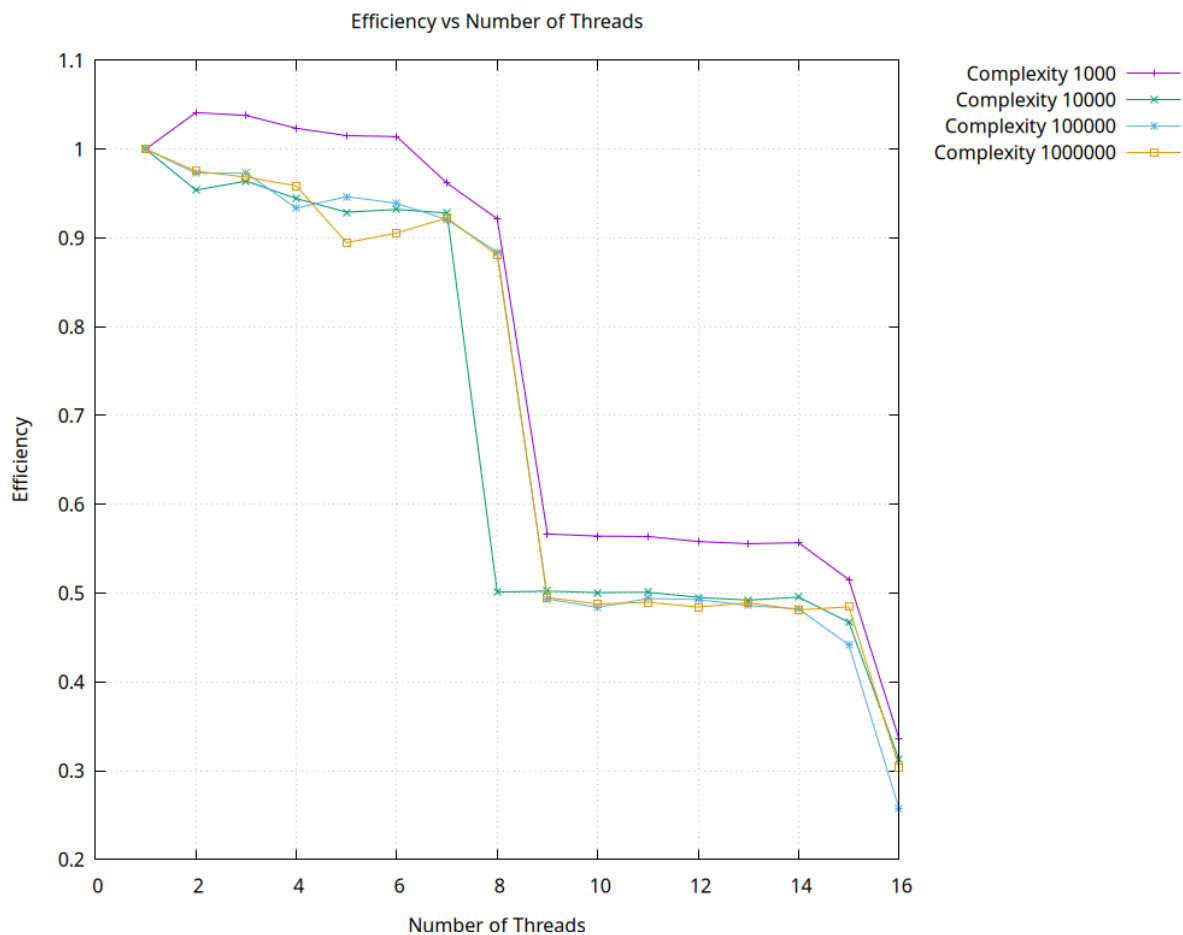


График 5. Эффективность от числа потоков и доп нагрузки

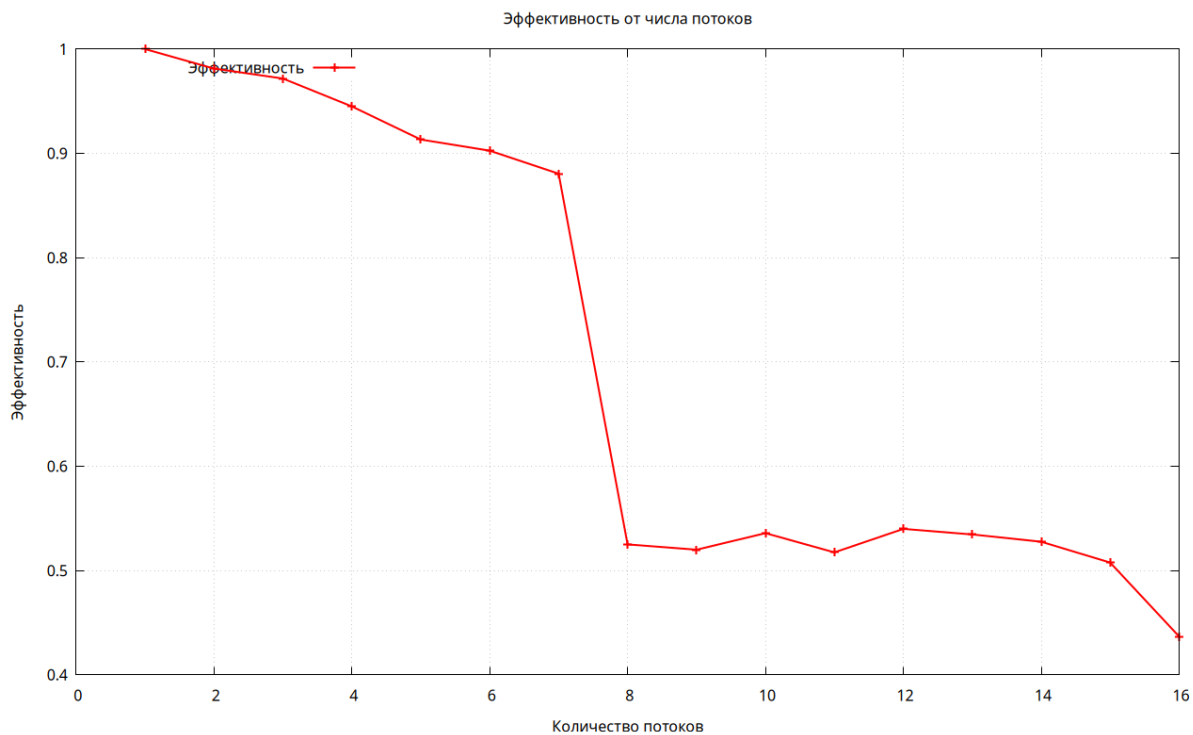


График 6. Эффективность от числа потоков

Анализ производительности программы для различных уровней сложности и без дополнительных вычислений:

1. Для уровня сложности 1000:

- С увеличением количества потоков до 6, ускорение и эффективность растут практически линейно, достигая пика эффективности около 1.014.
- После 7 потоков эффективность начинает падать, что связано с увеличением накладных расходов на управление потоками и синхронизацию. Например, при 9 потоках эффективность снижается до 0.566, а при 16 потоках до 0.335.
- Это указывает на наличие эффекта насыщения, когда добавление дополнительных потоков не приводит к улучшению производительности, а скорее снижает её из-за накладных расходов.

2. Для уровня сложности 10 000:

- До 7 потоков наблюдается стабильный рост ускорения и эффективности, хотя эффективность немного ниже, чем на уровне сложности 1000, с максимальной эффективностью около 0.932 при 6 потоках.
- Начиная с 8 потоков, эффективность резко снижается (например, 0.501 при 8 потоках), что подтверждает тот же эффект насыщения.
- При 16 потоках эффективность падает до 0.312, что свидетельствует о низкой эффективности работы программы при увеличении потоков свыше 8.

3. Для уровня сложности 100 000:

- Ситуация аналогична предыдущим уровням сложности: до 7 потоков программа показывает хорошее ускорение с максимальной эффективностью 0.938 при 6 потоках.
- Однако, начиная с 8 потоков, эффективность снова падает, что подтверждает ограничение по количеству полезных потоков.
- При 16 потоках эффективность составляет всего 0.257, что указывает на значительное падение производительности из-за накладных расходов.

4. Для уровня сложности 1 000 000:

- До 7 потоков наблюдается устойчивый рост производительности, с максимальной эффективностью 0.922 при 7 потоках.
- После 7 потоков эффективность также начинает падать, но здесь она стабилизируется на уровне около 0.484 для 15 потоков, что выше, чем на предыдущих уровнях сложности.
- Это указывает на то, что с увеличением сложности программа более эффективно использует большее количество потоков.

5. Для программы без дополнительных вычислений:

- Производительность демонстрирует хорошую масштабируемость до 7 потоков с эффективностью 0.880.
- После 7 потоков также наблюдается падение эффективности, но снижение не такое резкое, как при дополнительных вычислениях, достигая 0.436 при 16 потоках.
- Это говорит о том, что программа без дополнительных вычислений меньше страдает от накладных расходов и лучше масштабируется при увеличении потоков.

Заключение:

В оригинальном отчете, не было сказано, что работа выполнялась на ноутбуке, не подключенном к электросети, в результате чего показатели, были немного занижены, однако это не так сильно повлияло на вид зависимостей

1. **Линейное увеличение производительности** достигается только до 6-7 потоков, после чего начинают преобладать накладные расходы на управление потоками и синхронизацию, что снижает общую эффективность.
2. **Эффективность программы уменьшается при увеличении сложности вычислений**, так как большие объемы данных и более сложные операции требуют более длительной обработки на каждом потоке.
3. **Для оптимального использования ресурсов** рекомендуется использовать не более 6-7 потоков, так как дальнейшее увеличение количества потоков приводит к снижению производительности из-за накладных расходов.
4. Программа без дополнительных вычислений лучше масштабируется на большее количество потоков, чем программа с дополнительными вычислениями, что указывает на то, что накладные расходы на дополнительные вычисления значительно влияют на эффективность.

Таким образом, для достижения наилучших результатов в программах с многопоточностью важно учитывать не только количество потоков, но и сложность операций, чтобы избежать эффекта насыщения и потери производительности.