

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительных систем

ОТЧЁТ
по лабораторной работе №3
по дисциплине «Параллельные вычислительные технологии»

«Редукция и префиксная сумма»

Вариант №1

Выполнил:
студент группы ИУ-423
Горлов Р.В.

Проверил:
к.т.н. Пазников А.А.

Новосибирск – 2016

Введение

POSIX Threads — стандарт реализации потоков (нитей) выполнения. Стандарт *POSIX.1c, Threads extensions (IEEE Std 1003.1c-1995)* определяет API для управления потоками, их синхронизации и планирования.

Реализации данного API существуют для большого числа UNIX-подобных ОС (GNU/Linux, Solaris, FreeBSD, OpenBSD, NetBSD, OS X), а также для Microsoft Windows и других ОС.

Библиотеки, реализующие этот стандарт (и функции этого стандарта), обычно называются **std::thread**.

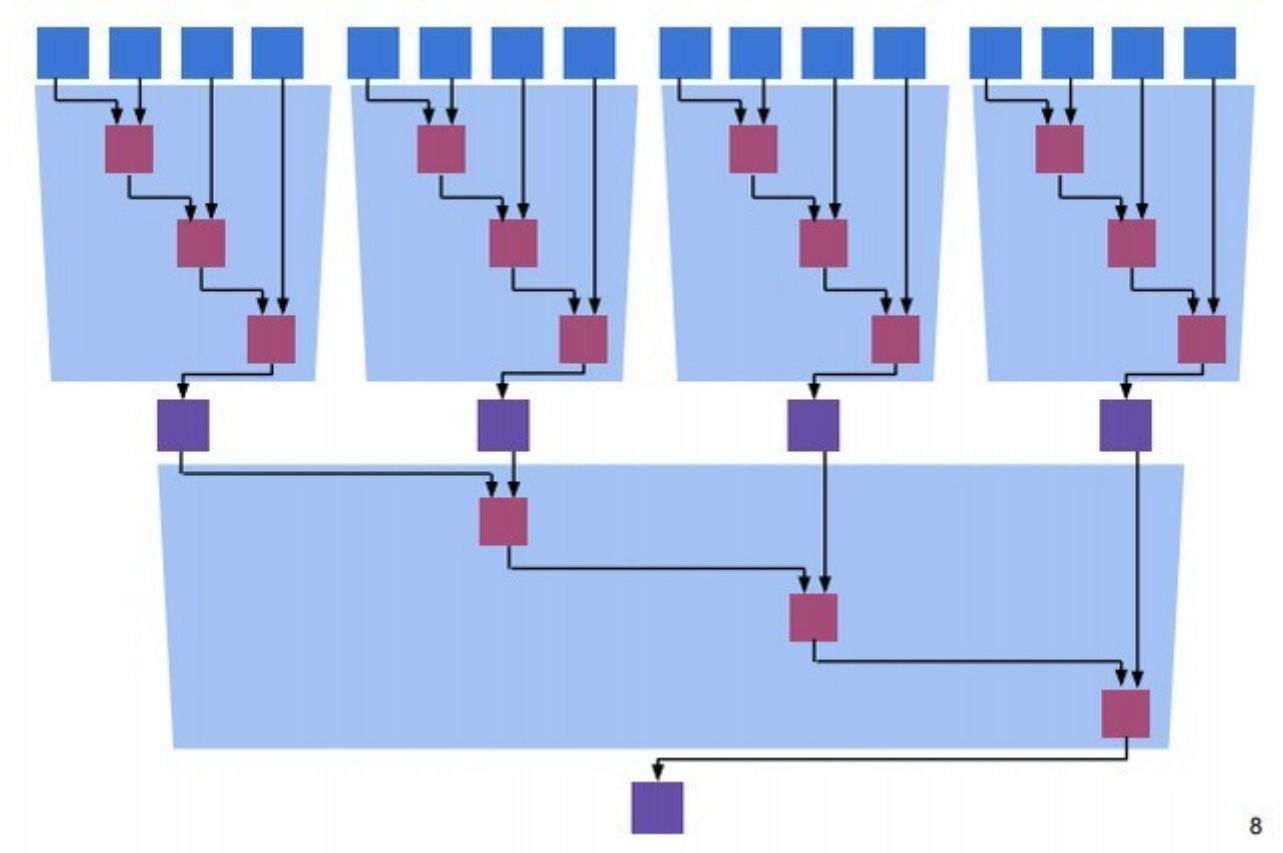
В данном варианте лабораторной работы при помощи библиотек POSIX std::threads на языке C/C++ в среде программирования QT был реализован алгоритм для операции редукции заданной ассоциативной операции элементов массива.

Потокобезопасность структуры данных проверяется за счёт использования свободно распространяемого инструмента Valgrind. Который позволяет проверять наличие гонок данных и других ошибок в многопоточных приложениях.

Описание

Редукция - ситуация, когда переменная внутри цикла используется для какой либо операции со значениями, полученных в каждой итерации.

Например, это происходит в цикле, который суммирует результаты вычислений, чтобы получить одно итоговое значение.



Реализация алгоритма

На вход функции которая запускает редукцию поступает структура содержащая:

- Вид операции;
- Массив с элементами;
- Количество элементов в массиве.

Локальные копии инициализируются соответственно типу оператора. Над локальными копиями переменных (или по указателю) после выполнения всех параллельной части алгоритма выполняется заданный оператор.

На выходе функции получаем значение — результат выполнения операции редукции для данной операции.

Анализ алгоритма

1. Коэффициент ускорения:

$$S_p = \frac{T_1}{T_p}$$

T_1 – время выполнения последовательного алгоритма;

T_p – время выполнения параллельного алгоритма на p процессорах;

S_p – ускорение на p потоков

2. Эффективность распараллеливания:

$$E_p = \frac{S_p}{p}$$

3. Закон Амдала:

Пусть программа содержит часть r , которая может быть распараллелена идеально (ускорение равно p) и полностью последовательную часть $s = 1 - r$, которая вообще не может быть распараллелена. Для каждого фиксированного n , максимально допустимое ускорение – $1 / s$.

Семейство графиков ускорения при разном объёме входных данных

Таблица 1: анализ программы для массива 1000000 элементов

Кол-во потоков	T_p	S_p	E_p
1	46ms	1	1
2	6ms	7	3,83
4	4ms	11	2,875
8	3ms	15	1,916

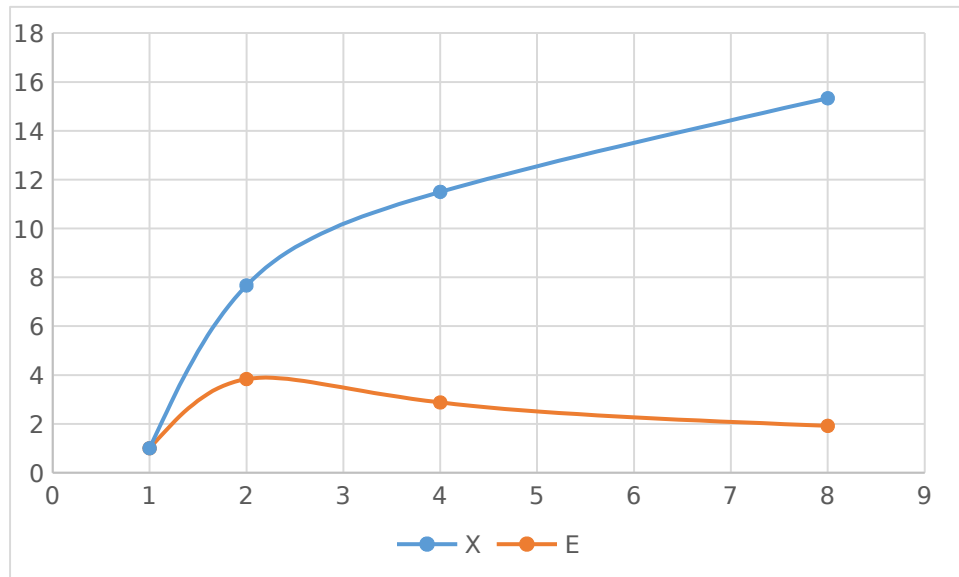
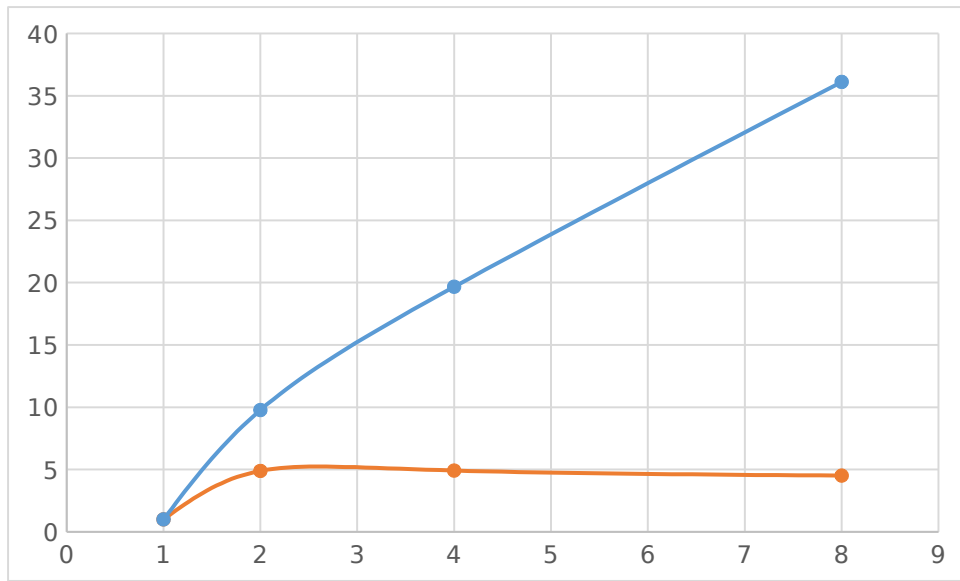


Таблица 2: анализ программы для массива 100000000 элементов

Кол-во потоков	T_p	S_p	E_p
1	4839ms	1	1
2	495ms	9	4,88
4	246ms	19	4,91
8	134ms	36	4,51



Вывод

В ходе выполнения данной лабораторной работы был реализован, протестирован и проанализирован алгоритм редукции, с использованием языка C/C++ и стандартной библиотекой «thread».