

## Эффективность многопоточных приложений Кадомский А. А.<sup>1</sup>, Захаров В. А.<sup>2</sup>

<sup>1</sup>Кадомский Андрей Андреевич / Kadomskij Andrej Andreevich – студент;

<sup>2</sup>Захаров Виктор Альбертович / Zaharov Viktor Albertovich – старший преподаватель,  
кафедра машиностроения и информационных технологий,

Зеленодольский институт машиностроения и информационных технологий (филиал),  
Казанский государственный технический университет им. А.Н. Туполева, г. Зеленодольск

**Аннотация:** в статье анализируется влияние многопоточности на скорость работы программы на компьютере с многоядерным процессором. Пример обеих программ (однопоточной и многопоточной) был выполнен на языке программирования С#.

**Ключевые слова:** потоки, процессы, многопоточность, эффективность.

Процесс – это программа (приложение), представленная в операционной системе как задача и управляемая супервизором задач. Для выполнения операционная система выделяет процессу необходимые ресурсы, в частности, область памяти, называемую адресным пространством процесса.

Под потоком понимают выполняющийся программный код.

Выполнение процесса начинается с выполнения основного потока, представленного главной функцией main или WinMain в языках C/C++ и Main() на языке С#. Основной поток инициирует, выполняет и завершает работу процесса [1, с. 160].

Из основного потока могут быть запущены дополнительные (рабочие) потоки, которые будут выполняться параллельно с основным и другими потоками.

Поток, в отличие от процесса, не имеет собственных ресурсов, поэтому все запущенные потоки будут выполняться в одном и том же адресном пространстве своего процесса.

Итак, если в программе запускаются дополнительно к основному потоку ещё и рабочие потоки, то приложение становится многопоточным. Возникает вопрос об эффективности многопоточных приложений относительно однопоточных.

Для примера был взят интеграл:

$$\int_a^b \left( \cos \sqrt{\sqrt{|\sin x + \cos x|}} \right)^{1.31} dx,$$

который многократно численно вычислялся. Вычисления выполнялись на компьютере с процессором Intel® Core™ i3-4170 CPU@3,7 GHz (2 ядра, 4 логических процессора). Тексты обеих программ (однопоточной и многопоточной) были написаны на языке С#. В многопоточной программе количество потоков было равно 4.

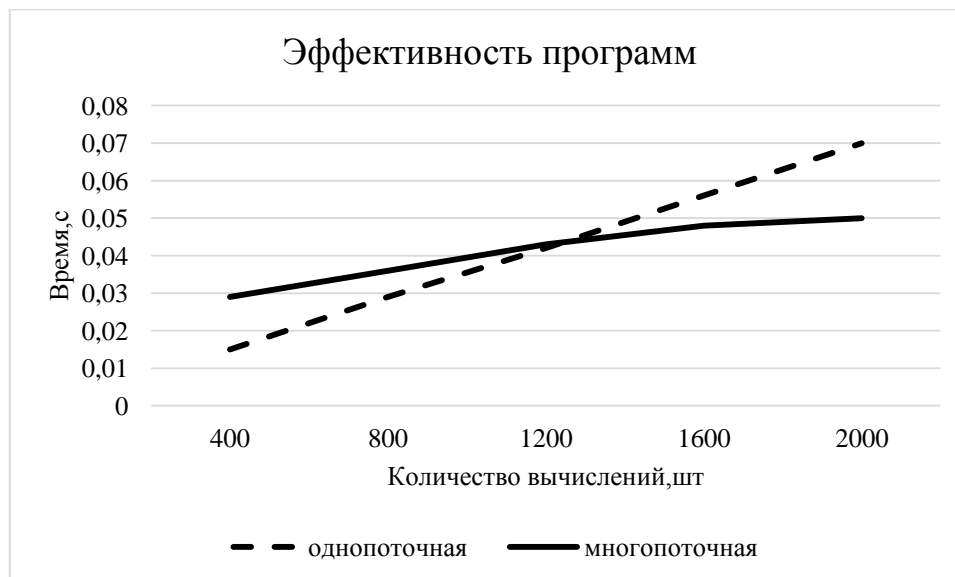


Рис. 1. Время работы однопоточных и многопоточных приложений при небольшом количестве вычислений

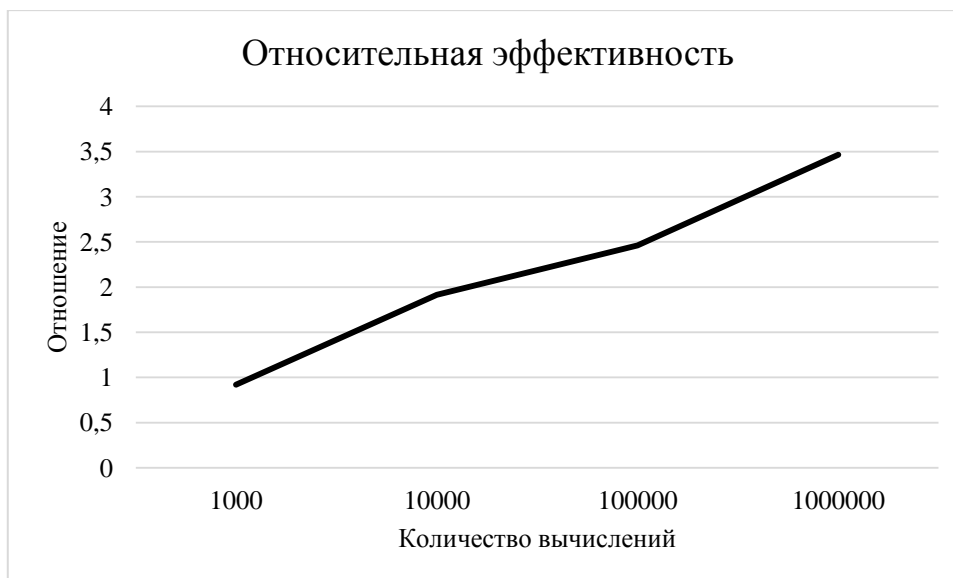


Рис. 2. Относительная эффективность многопоточных и однопоточных приложений

Как мы видим из графиков на рис. 1, многопоточный вариант программы при малом количестве вычислений может быть даже более медленным, чем однопоточный. Точка перехода для данного варианта приложения приблизительно равна 1200 вычислениям интеграла. С ростом количества вычислений эффективность многопоточного варианта относительно однопоточного возрастает. На рис. 2 приводится отношение времени выполнения многопоточного и однопоточного приложения при очень больших количествах вычислений. Как видно из данного рисунка, отношение времени выполнения стремится к количеству логических процессоров компьютера, что свидетельствует о высокой (почти 100%) загрузке многоядерного процессора при работе многопоточного приложения.

#### *Литература*

1. *Медведев В. И.* Особенности объектно-ориентированного программирования на C++/CLI, C# и Java. 2-е изд., испр. и доп. Казань: РИЦ «Школа», 2010. 444 с.