

Корначевський Я.І., Підоплічко В.С.  
ННК "ІПСА" НТУУ "КПІ"

## Дослідження ефективності виконання паралельних обчислень на різних технологіях

**1. Вступ.** Сьогодні зростання складності задач вже не компенсується нарощуванням частоти центрального процесора. Єдиний зараз вихід – розпаралелювання задач для виконання їх на кількох ядрах [1].

**2. Паралельне виконання.** У випадку роботи з кількома ядрами операційна система сама регулює розподілення задач між процесорами. На низькому рівні втручання в роботу цього механізму дуже складне і потребує неабияких знань у цій галузі. Прискорити виконання програми можна, використавши так звані потоки, які можна створювати з допомогою викликів до API системи [2].

Кожна мова програмування по-своєму організує створення таких потоків, тому постає запитання: яка з мов дозволяє мінімізувати накладні витрати на створення нового потоку-нитки і, таким чином, отримати максимальне прискорення?

Для того, щоб дати відповідь на це запитання, запропоновано дослідити ефективність паралельних обчислень, побудованих на технологіях FORTRAN, C, C++, .NET 2.0, JAVA, до того ж пропонується виконувати тестовий код на різних операційних системах.

Для початку проведено тестування ефективності виконання програми, написаної мовою C#, з використанням .NET [3]. Використовувався комп'ютер з процесором Intel CoreDuo з тактовою частотою 1,73 ГГц. Програма виконувала  $n$  операцій множення з плаваючою крапкою (кожна операція не залежить від результату виконання інших). При послідовному виконанні у циклі було затрачено деякий час  $t$ . Завантаженість процесора становила близько 50%. Причому, при зростанні завантаженості першого процесора, завантаженість на другому падала, і навпаки. Якщо розділити цикл на дві частини, кожна з яких реалізувати окремою функцією, а потім викликати їх послідовно в основній програмі, то час виконання зменшується на 10–20%. При використанні підходу з застосуванням потоків (клас Thread), робота була розподілена порівну між двома створеними потоками. В результаті час виконання становив  $0.6t$ – $0.7t$ . Використання класу ThreadPool дозволило досягти найбільшого прискорення. Час розрахунків становив відповідно  $0.4t$ – $0.45t$ .

**3. Висновки** Очевидно, що дані результати пояснюються в першу чергу особливостями реалізації віртуальної машини .NET, і, очевидно, що на швидкість безпосередньо впливає те, як програма працює з кеш-пам'яттю [1,3]. Зважаючи на ці неоднозначності, конче необхідно провести аналогічні тести з альтернативними технологіями, зокрема з C++.

## Бібліографія

1. Корначевський Я. І., Харченко К. В. Застосування бібліотеки .NET для паралельних розрахунків у багатопроцесорних системах, САІТ 2007.
2. Multi-core programming. Increasing performance through software multithreading, Shameem Akhter and Jason Roberts, Intel press, 2005.
3. C# 2005 для професіоналів.: Пер. с англ. – М.:Издательский дом "Вильямс", 2006. – 1376 с.: ил. – Парал. тит. англ.