

Корначевський Я.І., Харченко К.В.
НТУУ «КПІ» ННК «ІПСА»

Застосування бібліотеки .NET для паралельних розрахунків у багатопроцесорних системах

Сучасні апаратні засоби передбачають зміну підходів до нарощування обчислювальних потужностей. Наприклад, подальше зростання робочих частот і мініатюризація наражається на елементарні фізичні обмеження. Тому одним з очевидних способів прискорення розрахунків є побудова обчислювальної системи на основі кількох ядер, що, очевидно, має гарантувати збільшення швидкості пропорційно до кількості ядер. Однак переважна більшість існуючого програмного забезпечення створювалася в рамках ідеології виконання на одному ядрі. Тому існує необхідність доопрацювання існуючого програмного забезпечення з точки зору виконання його на кількох ядрах. Дана робота присвячена дослідженню особливостей застосування бібліотеки .NET при роботі в багатопроцесорній системі.

Кожна операційна система по-своєму забезпечує паралельне виконання програм і потоків в межах існуючих програм. Для цього використовується так званий «Context Switch», з допомогою якого зберігається інформація про поточний стан процесора і кожен потік і, таким чином, не відбувається накладання даних. В .NET для роботи з потоками використовуються засоби CLR (common language runtime) і бібліотека System.Threading, клас Thread. При створенні класу потоку йому передається так званий delegate, який є еквівалентом функції в C++.

Тестовою задачею обрано операції по роботі з векторами та матрицями. Для цього розроблено набір тестів, які дозволяють виконувати розрахунки із застосуванням різної кількості паралельних потоків, незалежно від кількості ядер виконавчої системи. Проведені тести показали, що паралельне виконання правильно організованої процедури завжди дає прискорення, практично пропорційно кількості процесорів. Однак закон Амдала може виконуватися не точно. Це пояснюється особливостями роботи системи з кеш-пам'яттю. Якщо забезпечити локалізацію операцій з однією (кількома) змінною в межах одного циклу, то, очевидно, що робота з нею буде виконуватися тільки з використанням швидкої кеш-пам'яті, що веде до додаткового зростання прискорення.

Бібліографія

- [1] Multi-core programming. Increasing performance through software multithreading, Shameem Akhter and Jason Roberts, Intel press, 2005.
- [2] C# для професіоналів.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1376 с.: ил.
- [3] Y.Serdyuk, «MC# 2.0: a language for concurrent distributed programming based on .NET», .NET Technologies 2006, 4th International Conference, University of West Bohemia, Plzen, Czech Republic, May 29 – June 1, 2006, <http://dotnet.zcu.cz>