

УДК 621.855

В.А. Лабчук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ ПОКРАЩЕННЯ ШВИДКОДІ ОБРОБКИ ДАНИХ У PLINQ В ПОРІВНЯННІ З LINQ ДЛЯ РІЗНИХ РОЗМІРІВ ВХІДНИХ ДАНИХ

V.A. Labchuk

RESEARCH OF THE DATA PROCESSING PERFORMANCE IMPROVEMENT USING PLINQ IN COMPARISON WITH LINQ ON DIFFERENT SIZED DATASETS

Під час консолідації даних з різних ресурсів ми зазвичай отримуємо дані великих обсягів. Щоб продуктивність їх обробки була високою, доцільно використовувати не лише LINQ, але й його паралельну версію – PLINQ. Однак чи в усіх випадках PLINQ буде продуктивніше ніж LINQ? Чи можливі ситуації коли PLINQ взагалі програватиме у швидкості? Коротка відповідь – так.

Можуть бути різні причини, коли паралельна обробка LINQ не буде виконуватися швидше ніж послідовна, зокрема розмір вибірки. Розмір вибірки є ключовим параметром і безпосередньо впливає на швидкість обробки. Звісно, чим більший розмір наших даних, тим довше їх потрібно обробляти. Однак саме в таких випадках спостерігається користь від паралельної обробки. Якщо ж даних у нас мало, то PLINQ не здатний забезпечити перевагу у швидкості, а деколи навіть навпаки потребуватиме додаткового часу на розпаралелювання, а також синхронізацію потоків, яка непотрібна при послідовному виконанні. Тобто якщо ми знаємо, що набір даних передбачається невеликий, використання PLINQ буде лише на шкоду.

Однак позитивним фактором від використання PLINQ саме при консолідації є те, як зростає час обробки при збільшенні розміру вибірки. Під час використання LINQ час збільшується експоненційноподібно, і вже на вибірках середнього розміру починає перебільшувати відповідний час при використанні LINQ. Використання ж PLINQ дозволяє часу збільшуватися більш лінійно при збільшенні обсягу вибірки. Дослідимо залежність часу знаходження простих чисел (рис. 1) від розміру вибірки, тобто діапазону в якому ми шукаємо ці прості числа [1].

Висновок. Досліджено залежність швидкості обробки та пошуку даних залежно від розміру вхідних даних. Зокрема досліджено час пошуку простих чисел на різних діапазонах для PLINQ та LINQ. Отримані результати показують велику вигоду від використання паралельної версії LINQ на великих вхідних масивах, які зазвичай і виникають при консолідації даних, та беззмисловність її використання для вхідних вибірок невеликих розмірів.

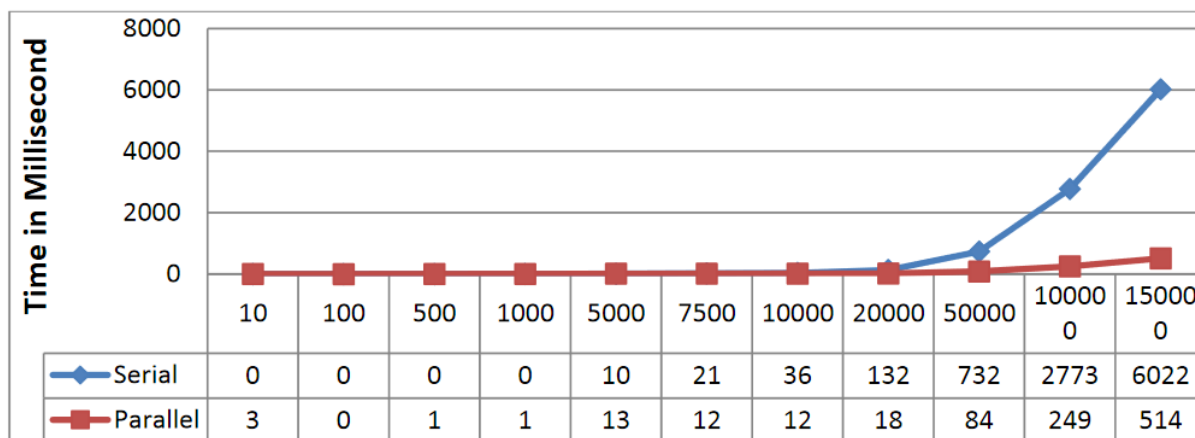


Рисунок 1. Порівняння паралельної та послідовної обробки даних LINQ та PLINQ на різних розмірах вхідних даних

Література

1. Md. Rashedul Islam, Alope Kumar Shah, Md. Rofiqul Islam. Experimental evaluation of parallelism in real time execution. No 36, P.49-51. Academia edu. https://d1wqtxts1xzle7.cloudfront.net/32324848/RNIS114PPL-libre.pdf?1391541786=&response-content-disposition=inline%3B+filename%3DExperimental_Evaluation_of_parallelism_i.pdf&Expires=1733085003&Signature=T2Xh9IZd7EQYufmU~i0Uz1XfDVHfD53xp2gSP~GrOzrDqNxtKB0Ex0J6qIfEHgcx7pRjiXSBtTCdC5ekSsRtHgS3LrR-W15W0MRTDUe4Jv~2eejLdEmKYqtWdZCom5c3y5kPfejwQX5ZlhPB3HRR7cghk4xdrEA6-EFZyCpU2lwMOV~~w32ZYd4J0c7Odun8wgLmaFSkf20B2q6aUPJpLN0mVdBmDH7brNPvbIF44PSqn5ldU9r~5emEwqTZ7Oe~DfUemlV~gdFaqJarmOjaPC8E2eS4GHD5cIGHfZns4DxHqYBdr6RHws8dY~hgs6W-P4cZUomLxuzf1Y14u87P5g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA