Два головні зауваження щодо даного завдання. Перше — оскільки звернення до матриць A і B виконується тільки для читання, а запис елементів матриці C йде незалежно один від одного, умов гонки не виникає, і жодна синхронізація не потрібна.

Друге — розподіл роботи між потоками може виконуватися з використанням «диспетчера», до якого потоки звертаються за завданням, але цей варіант складніший і потребує більших накладних витрат, що може погано відбитися на ефективності. Тому було прийнято рішення розподілити роботу між потоками порівну відразу при їх створенні. Для цього я скористався тим, що всі елементи матриці C можна перерахувати по рядкам, так що елементу C[n][k] відповідатиме номер j = n*K+k, і, відповідно, індекси елементу C[n][k] обчислюються як n = j/K, k = j%K. Таким чином кожен потік (крім, можливо, останнього) отримує приблизно однакову кількість елементів для обчислення — (N*K+tCount-1)/tCount, де tCount — кількість потоків.

Завдання 1.1. *(2 бали) Продемонструвати паралелізм (непослідовність) обчислень через виведення результату (трійками [x,y]=result) “по ходу обчислень”.*

Результат продемонстровано на наведеній копії екрану. Числа ліворуч — ідентифікатори потоків. Різнокольоровий вивід дає змогу наочно побачити багатопоточність програми.

Завдання *1.2.* (+3 бали) Дослідити швидкодію A*B залежно від кількості потоків для розпаралелення множення. Продемонструвати та пояснити цю залежність. За якої кількості потоків множення буде найшвидшим? Підтвердити експериментально.*

Результати замірів часу для множення матриць розміру A[200][10000]×B[10000][200] наведені на рис. 1.

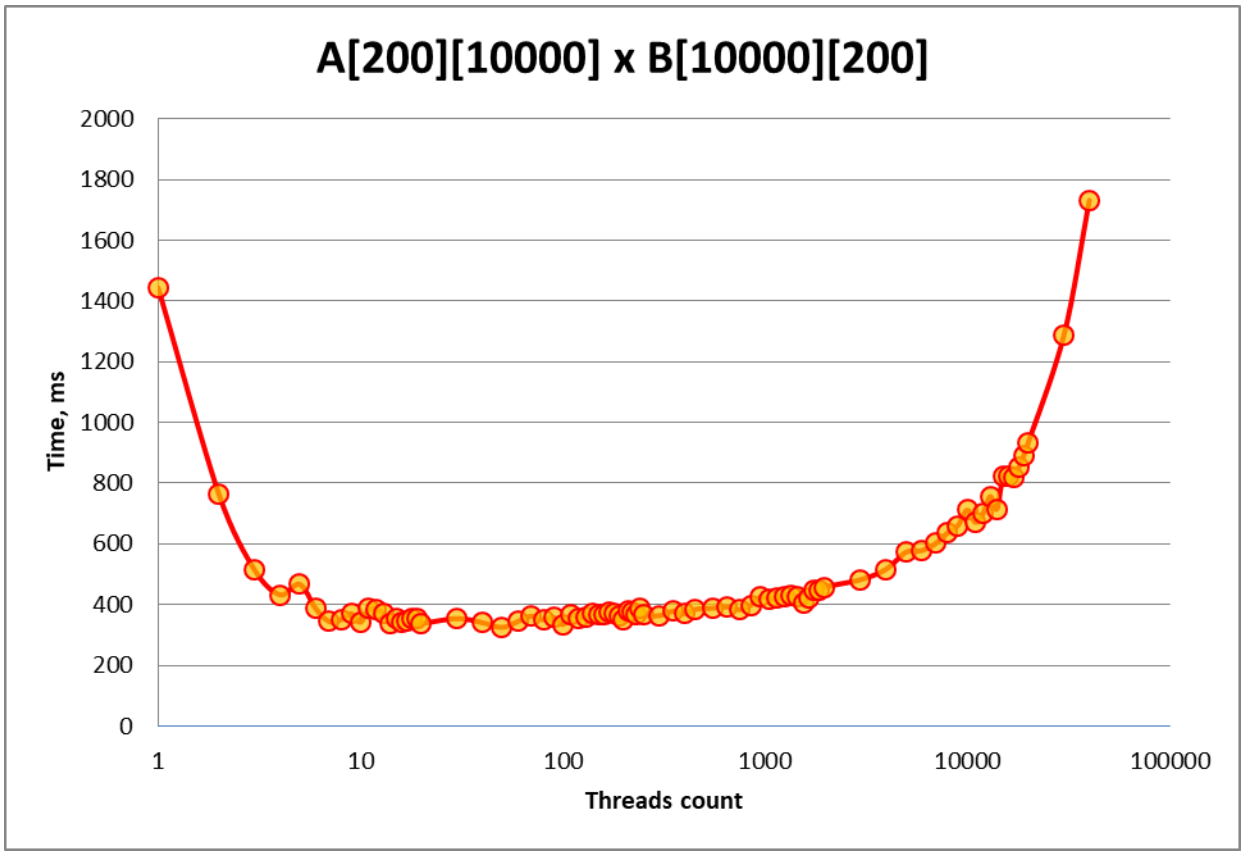

Рис. 1. Час обчислення перемноження матриць A[200][10000]×B[10000][200] залежно від кількості потоків

Даний комп'ютер має процесор Intel(R) Core(TM) i7-4790 CPU@3.60GHz з кількістю фізичних ядер 4 (кількість логічних процесорів 8).

Дану залежність можна пояснити тим, що коли кількість потоків менша за кількість логічних процесорів, кожен потік виконується на своєму логічному процесорі без перемикань, і швидкість обчислень зростає.

Але з подальшим ростом кількості потоків приросту швидкості немає, бо скільки б потоків не було запитано, одночасно можуть виконуватися лише 8 потоків. При цьому, оскільки операційна система в такому випадку вимушена перемикати потоки, на таке перемикання потоків витрачається процесорний час, який забирається в основних обчислень, тож загальний час виконання обчислень зростає. Отже, використовувати більшу кількість

активних обчислювальних потоків, ніж кількість логічних процесорів, немає жодного сенсу.

Невеликі стрибки на графіку можна пояснити різними побічними ефектами, як-то використання кешу процесора або фонове виконання задач самої операційної системи, для якого теж потрібен процесорний час.

Результати роботи для інших розмірів матриць демонструють залежності такого самого типу.