Аналіз часу роботи блочного сортування в одно- та багато-поточному випадках

Конфігурація обладнання (елементи що впливають на швидкість):

* Процесор: AMD FX-8320 Eight Core processor
* Пам’ять: DDR3 1866MHz

Дана багато-поточна реалізація блочного сортування, відносить один потік до кожного блоку. Тобто кожен блок обробляється 1-м потоком і у випадку подальшого рекурсивного блочного сортування створює нові потоки, що займає досить значний час. Відповідно при глибокій рекурсії час роботи даної програми значно зростає (порівняно з одно-поточною версією). Проте якщо глибина рекурсії невелика то багато-поточна версія сортування є значно ефективнішою.

# Одно-поточна версія:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Максимальний розмір блоку для рекурсивного блочного сортування | Кількість елементів масиву: | 1000 | 10000 | 100000 |
| 5 | | 1ms | 8ms | 84ms |
| 10 | | 1ms | 7ms | 67ms |
| 20 | | < 1ms | 6ms | 64ms |
| 25 | | 6ms | 61ms |
| 50 | | 5ms | 50ms |
| 100 | | 5ms | 48ms |
| 200 | | 5ms | 48ms |
| 400 | | 5ms | 50ms |
| 800 | | 5ms | 51ms |
| 1600 | | 5ms | 55ms |
| 3200 | | 6ms | 56ms |
| 4800 | | 6ms | 65ms |
| 6400 | | 6ms | 64ms |

# Багато-поточна версія:

«-» - переповнення стеку (завершення роботи програми).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Максимальний розмір блоку для рекурсивного блочного сортування | Кількість елементів масиву: | 1000 | 10000 | 100000 |
| 5 | | 213ms | 1569ms | - |
| 10 | | 104ms | 1108ms | - |
| 20 | | 58ms | 357ms | - |
| 25 | | 57ms | 255ms | - |
| 50 | | 21ms | 248ms | 1173ms |
| 100 | | 14ms | 57ms | 1174ms |
| 200 | | 10ms | 56ms | 252ms |
| 400 | | 7ms | 34ms | 256ms |
| 800 | | 7ms | 15ms | 171ms |
| 1600 | | 7ms | 14ms | 66ms |
| 3200 | | 7ms | 8ms | 64ms |
| 4800 | | 7ms | 8ms | 36ms |
| 6400 | | 7ms | 9ms | 36ms |

Таким чином можна зробити висновок, що через певні обмеження, такі як:

* Створення потоку займає час.
* Для коректної роботи необхідна деяка синхронізація потоків (тобто деякий час цей потік «простоює», але споживає ресурси комп’ютера).

багато-поточна реалізація блочного сортування в деяких випадках не є швидшою за одно-поточну.

Щоб уникнути вищевказаних проблем, можна використати, наприклад Thread Pool.