

Párhuzamos Algoritmusok

Buborékredezés

Kórád György (ZF440N)

2023. május 12.

Tartalomjegyzék





0.1. A feladat leírása	2
0.2. Kapott eredmények	3
0.3. Konklúzió	4

0.1. A feladat leírása

A feladat célja egy 100000 elemű tömb rendezése szekvenciálisan, majd szákezeléssel. Végül a kapott eredményeket összehasonlítjuk és az eredmények alapján eldöntjük, melyik módszer a leggyorsabb. A buborékredezés egy egyszerű algoritmus, hatékonysága rosszabb, mint a más összehasonlításos rendezési algoritmusoké. Az átlagos és legrosszabb esetben is $O(n^2)$ időkomplexitással rendelkezik, ahol "n" a tömb mérete.

```
void bubbleSort(int arr[])
{
    for (int i = 0; i < N ; i++)
    {
        for (int j = 0; j < N - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

Buborékredező algoritmus

Hardware Information	
Aspire E5-573G V3.72	
	Intel i5-4210U (4) @ 2.7GHz
	Intel Haswell-ULT
	NVIDIA GeForce 940M
	1974MiB / 7866MiB (25%)

A tesztek futtatásához használt hardware

Az elkészült programok indulást követően bekérlik a felhasználótól a tömb méretét, és a szálak számát, így különböző problémamérettel is lehet vizsgálatokat végezni. Az OpenMPI-vel készült program csak a tömb méretét kéri, mivel itt paraméterként kell megadni a szálak számát a programnak.

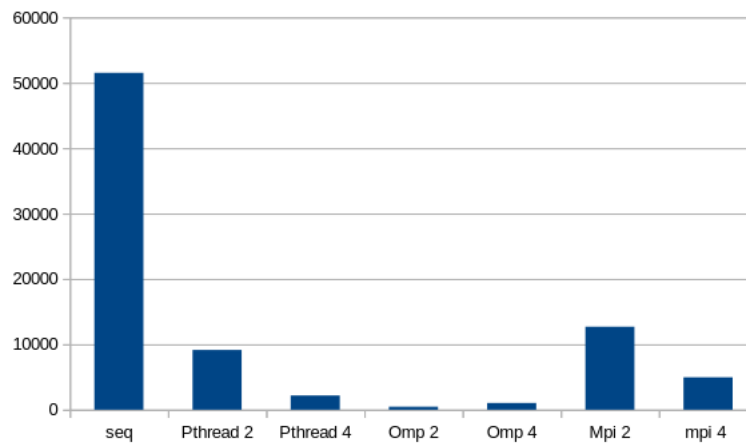
0.2. Kapott eredmények

Az algoritmust a szálkezelő technológiák használatával kettő illetve négy szálon futtattam. Majd az így kapott eredményeket egy file-ban tároltam.

Sequential	51512 ms
Pthread 2	9134 ms
Pthread 4	2165 ms
Omp 2	447 ms
Omp 4	1020 ms
Mpi 2	12687 ms
Mpi 4	4947 ms

Futási idők

Érdekes módon az OpenMP technológia két szálon gyorsabb volt, mint négyen, viszont még így is magasan a leggyorsabb mind közül.



Futási idők diagramja

0.3. Konklúzió

TECH	MS	Acceleration
Sequential	51512	0
Pthread 2	9134	463,96%
Pthread 4	2165	2279,31%
Omp 2	447	11423,94%
Omp 4	1020	4950,20%
Mpi 2	12687	306,02%
Mpi 4	4947	941,28%

Eredmények javulása százalékban

Összességében elmondható, hogy az OpenMP volt a leggyorsabb, és véleményem szerint a használata is ennek volt a legegyszerűbb.

A Message Passing Interface (MPI) is látható gyorsulást mutat több szál esetében, viszont ennél érdemes megjegyezni, hogy nem a program forráskódjában adjuk meg a szálak számát, hanem futtatási paraméterként.

```
mpirun --oversubscribe <szálak> <program>
```

Harmadik helyen végzett a pthread-et alkalmazó program, ennél a szálak számát növelve egyre jobb eredményeket kaptam, viszont a szálak egy bizonyos száma után romlottak a kapott eredmények.