

Descriere scurta a caracteristicilor procesorului pe care s-au efectuat experiment



Base speed:	2.38 GHz
Sockets:	1
Cores:	6
Logical processors:	6
Virtualization:	Enabled
L1 cache:	384 KB
L2 cache:	3.0 MB
L3 cache:	8.0 MB

Procesor: AMD Ryzen 5 4500U

Frecventa de baza: 2.38 GHz

Frecventa in turbo pana la: 4.0 GHz

Numar core-uri: 6

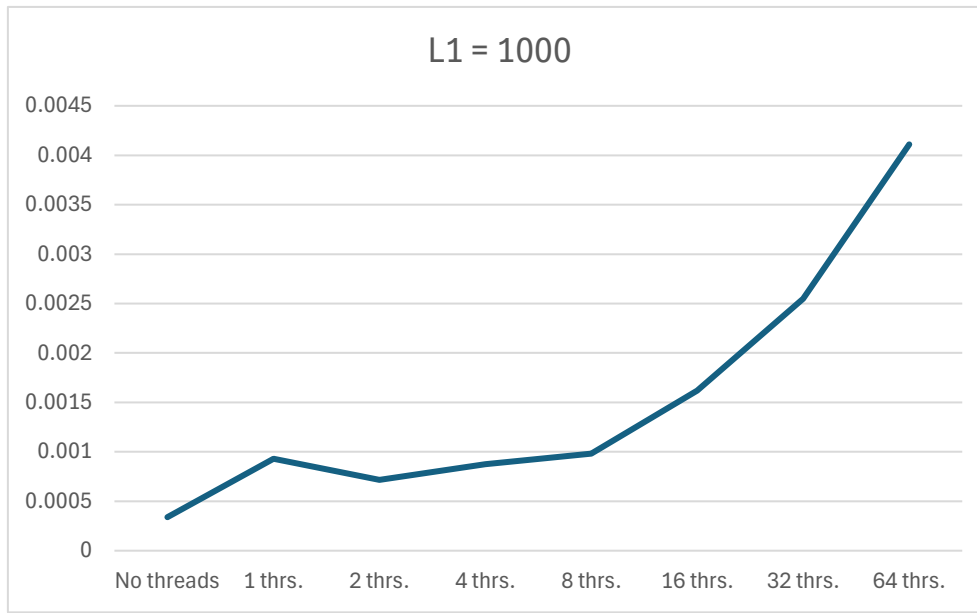
Numar threaduri-uri: 6

Instructiuni pe intregi: 31,409 Mops/Sec

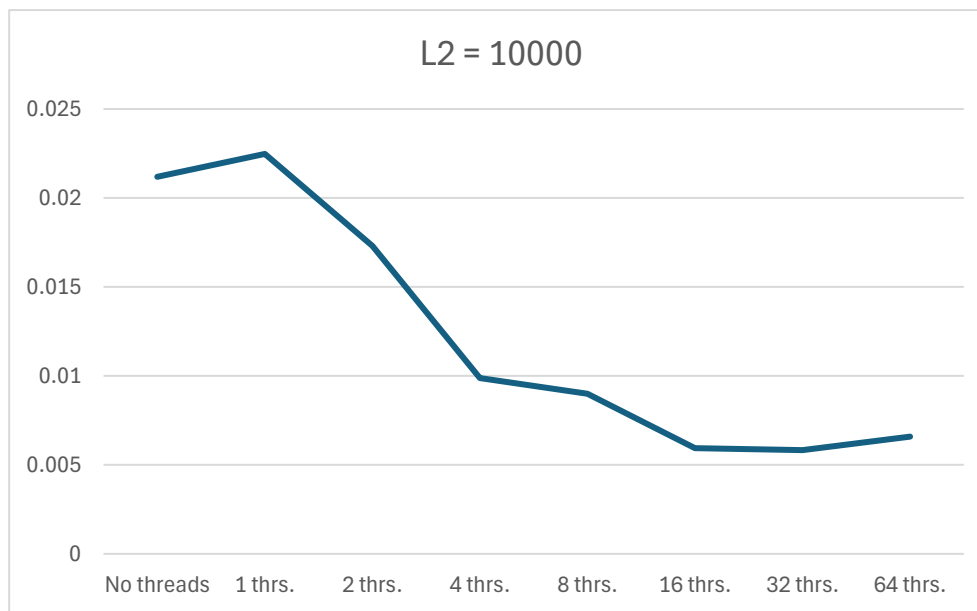
Instructiuni in virgula mobila: 22,132 Mops/Sec

Exercitiul 2 din laboratorul 1:

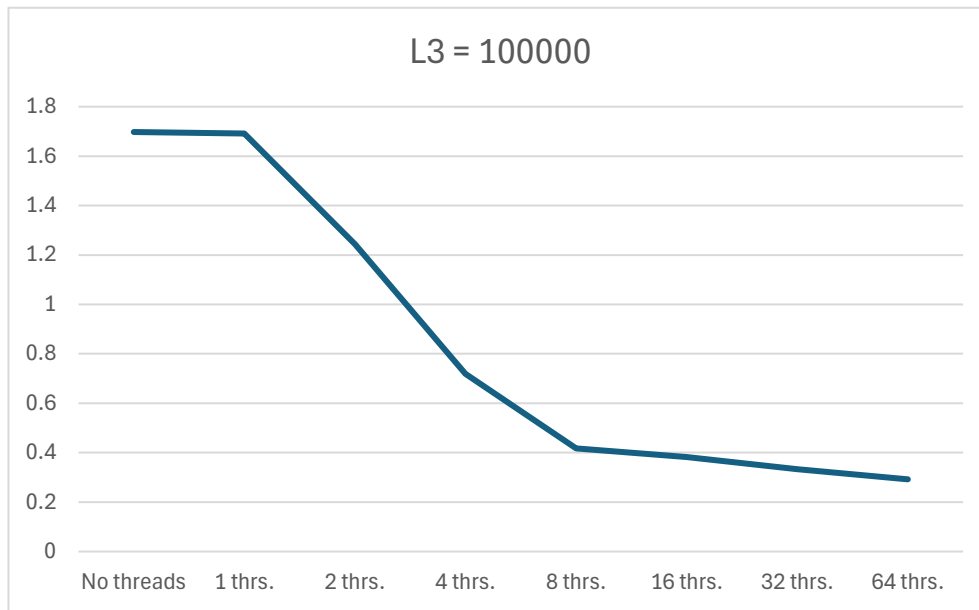
Upper limit value	Execution time (clock cycles)							
	No threads	1 thrs.	2 thrs.	4 thrs.	8 thrs.	16 thrs.	32 thrs.	64 thrs.
L1 = 1000	0.000339	0.000931	0.000717	0.000875	0.000982	0.001619	0.002547	0.00411
L2 = 10000	0.021196	0.022475	0.017304	0.009873	0.00899	0.005944	0.00583	0.00659
L3 = 100000	1.697369	1.690939	1.243113	0.719306	0.417751	0.382231	0.33331	0.292169
L4 = 1000000	137.4563	137.7352	101.5721	58.62837	34.23887	26.89843	24.40105	23.70688
L5 = 2000000	531.2873	526.9761	387.0823	223.3491	130.546	101.703	92.57709	90.53171



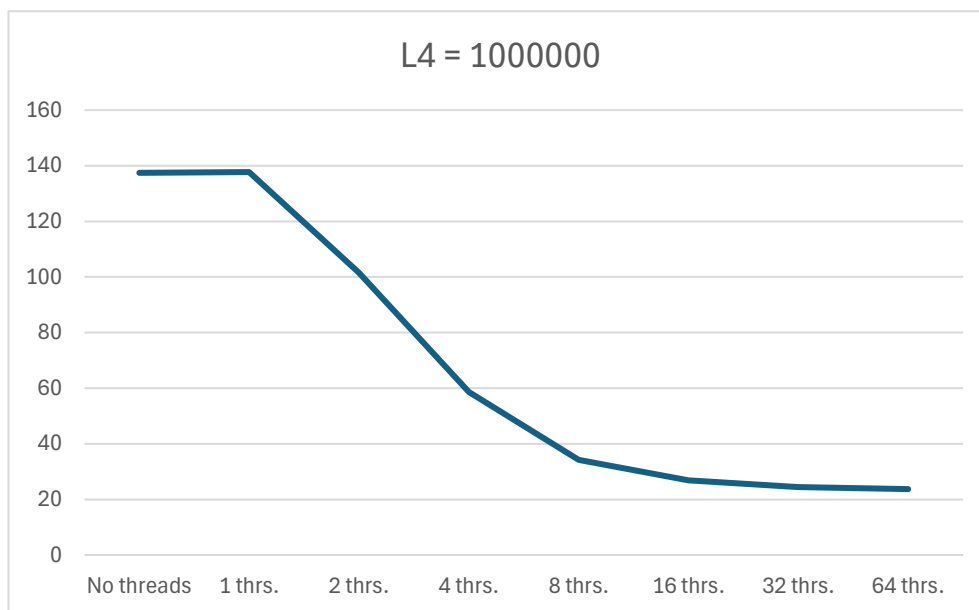
Pentru $L1 = 1000$, valoarea fiind mica timpul cel mai bun e obtinut pentru implementarea care nu foloseste thread-uri. In acest caz cu cat crestem numarul de thread-uri timpul de executie creste.



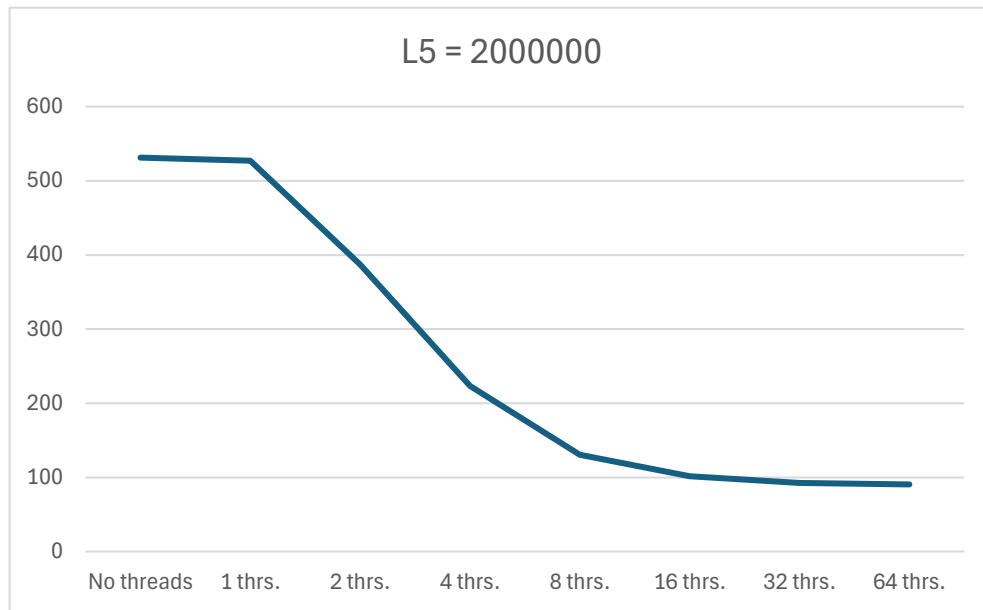
Pentru $L2 = 10000$, se observa ca odata cu cresterea numarului de threaduri timpul scade, dar valoarea fiind una nu foarte mare, nu mai apar imbunatatiri dupa ce se trece de pragul de 16 thread-uri.



Pentru L3 = 100000, se vad imbunatatiri de fiecare data cand se creste numarul de thread-uri, dar o data cu trecerea pragului de 8 threaduri imbunatatirile sunt foarte mici.



Pentru L4 = 1000000, din nou se observa imbunatatiri o data cu cresterea numarului de threaduri, dar din nou dupa trecerea pragului de 8 threaduri imbunatatirile sunt tot maici, iar diferenta dintre 32 si 64 de threaduri e



Pentru $L5 = 2000000$, rezultatele sunt aproximativ la fel ca pentru $L4$, dar deja chiar si diferenta dintre 16, 32 si 64 e foarte mica.

In concluzie, in urma testelor se observa ca folosirea unui singur thread nu face nici o diferenta fata de folosirea doar unui thread, iar folosirea mai multor threaduri e foarte convenabila pentru cand avem de calculat o multime de valori reducand cu mult timpii de executie. Totodata se observa ca dupa ce numarul de thread-uri depaseste numarul de core-uri ale procesorului, imbunatatirile timpilor devin tot mai mici odata cu cresterea numarului de threaduri.

Exercitiile 1 si 2 din laboratorul 2:

tele: