Софийски университет “св. Климент Охридски”
Факултет по математика и информатика

Курсов проект

По Системи за паралелна обработка

Тема - изобразаване на фрактал:Mandelbrot set

Изготвил: Александър Витков, ФН 81618

Проверил: .....

Задача (проект №18)

Да се имплементира паралелен алгоритъм, генериращ изображение на даден регион от множеството Манделброт - **M**, дефинирано като:

M е множеството от точки ***c*** в комплексната равнина, за безкрайната редица **Series(c)** е ограничена:

F(*Z*) := *c* * cos(*Z*)
Series(*c*) := { c_0 = 0+0i, c_1 = F(c_0), c_2 = F(c_1), c_3 = F(c_3), … }

Имплементация

Програмата имплементира SPMD алгоритъм със статично балансиране на натоварването, генериращ изображение на регион от комплексната равнина, в което белите пиксели са точките, които са част от **M**, а тъмно-оцветените не са част от **M**.

Следните стойности се приемат като командни аргументи:
--size или ***-s*** - размер на изображението, default 800x600
--rect или ***-r*** - регион от комплексната равнина, default -2:-2,-1:1
--tasks или ***-t*** - брой нишки, default 1
--granularity или ***-g*** - грануларност, default 1
--iter или **-i** - максимален брой итерации за всики пиксел, default 50
--output или ***-o*** - име на файл, в който да се запише изображението

За всеки пиксел се намира съответната точка от равнината **c** и се провежда цикъл, който пресмята първите ***iter*** елемента от **Series(c)**.
Използва се факта, че ако $\exists c_i \in$ **Series(c): $|c_i| > 2.0$**, то редицата **Series(c)** е разходяща - ако по време на цикъла достигнем число с модул по-голям от 2, то **c**$\notin$**M**.
Ако след ***iter*** итерации все още не сме открили такъв елемент, приемаме с достатъчно добра точност, че **c**$\in$**M**.

Модел на паралелизъм

За ускорение се се използва SPMD модел на паралелизъм.
Масивът с данни за изображението, което ще бъде генерирано се разбива на **chunks_count** парчета, където **chunks_count := *granularity* * *tasks***.

Създават се ***tasks*** на брой работни нишки номерирани **0..tasks-1**, всяка от които обработва статично предопределени парчета.

Балансиране на натоварването

Използва се статично балансиране на натоварването - определя се, че **n**-тата нишка ще обработи **chunks[i]**, когато **i % tasks == n**.

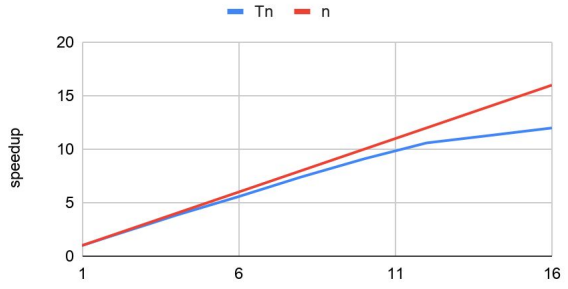
Ако например например имаме 8 нишки - номерирани **0..7** и зададена **грануларност 4**, то **chunks_count := 8*4 = 32**.
Нулевата нишка ще обработи парчета 0, 8, 16, 24, първата нишка ще обработи 1, 9, 17, 25, …, и седмата нишка ще обработи парчета 7, 15, 23, 31.

Тест за ускорение при различен брой нишки

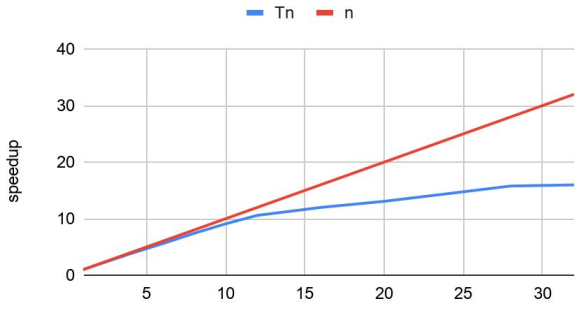
Всички тестове са проведени в сървъра t5600.rmi.yaht.net на 26 август по време на ужесточена борба за сървърни ресурси. Следният тест проверява ускорението при грануларност 16 с брой нишки от 1 до 32 върху default региона от равнината:

Image size 4000x4000, granularity 16, default region						
num threads	Tn - average of the two runs	run 1	run 2	expected speedup	Sn	En
1	39.076s	39.274s	38.878s	1	1	100%
2	20.180s	20.205s	20.155s	2	1.95	97.31%
3	13.634s	13.639s	13.628s	3	2.88	96.02%
4	10.267s	10.307s	10.226s	4	3.83	95.64%
6	7.046s	7.070s	7.021s	6	5.57	92.91%
8	5.298s	5.301s	5.295s	8	7.41	92.66%
10	4.316s	4.301s	4.331s	10	9.10	91.00%
12	3.707s	3.706s	3.708s	12	10.59	88.29%
14	3.480s	3.635s	3.324s	14	11.29	80.62%
16	3.275s	3.349s	3.200s	16	11.99	74.96%
20	3.004s	3.060s	2.948s	20	13.07	65.37%
24	2.724s	2.693s	2.754s	24	14.42	60.09%
28	2.489s	2.539s	2.438s	28	15.78	56.36%
32	2.458s	2.507s	2.409s	32	15.98	49.93%

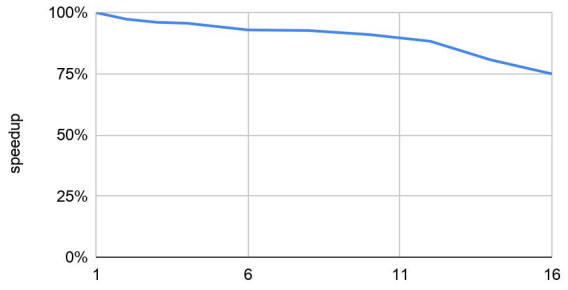
Ускорение, до 16 нишки



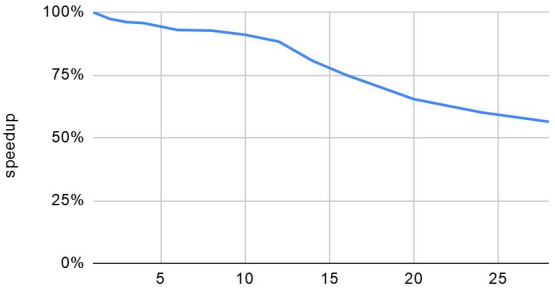
Ускорение, до 32 нишки



Ефективност - En, до 16 нишки



Ефективност - En, до 32 нишки

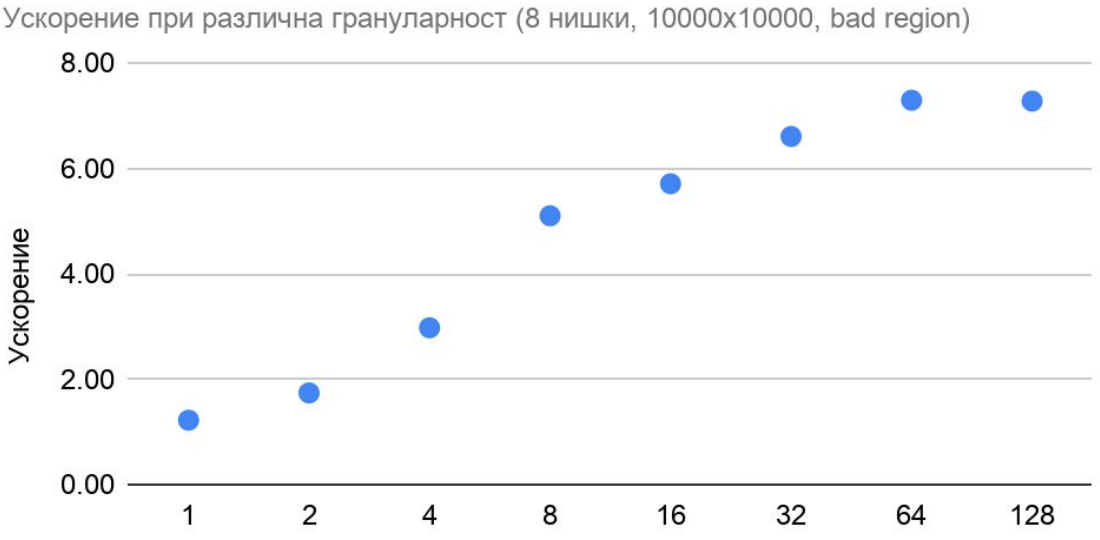


Тест за ускорение при различна грануларност

Следният тест е върху региона **-2:2,0:10**. Този регион е специално избран, защото голяма част от него е извън множеството и демонстрира лош случай за изпълненията с ниска грануларност.

Вижда се, че до определен момент, като повдигаме грануларността, разликата между най-бързата и най-бавната нишка се скъсява.

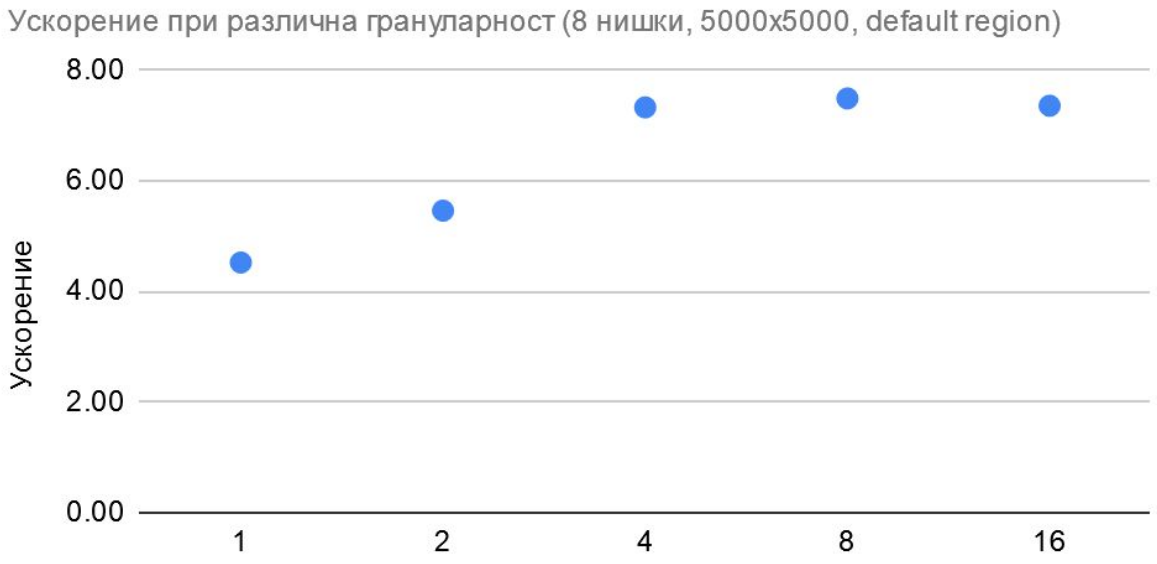
10000x10000, granularity test on bad region, 8 threads						
num threads	granularity	fastest thread	median	slowest	speedup	efficiency
1	1	30.863s	30.863s	30.863s	1	100%
8	1	0.661s	0.676s	25.204s	1.22	15.31%
8	2	0.659s	1.072s	17.710s	1.74	21.78%
8	4	0.669s	1.990s	10.358s	2.98	37.25%
8	8	0.901s	4.760s	6.044s	5.11	63.83%
8	16	2.819s	4.305s	5.401s	5.71	71.43%
8	32	3.539s	3.965s	4.667s	6.61	82.66%
8	64	3.708s	3.964s	4.227s	7.30	91.27%
8	128	3.869s	3.975s	4.237s	7.28	91.05%



Тест за ускорение при различна грануларност II

Следният тест е проведен върху региона по подразбиране (-2:2,-1:1). Понеже голяма част от региона е в множеството M, голяма част от циклите за се изпълняват до края за всеки пиксел, затова всички нишки са по натоварени и не е нужна толкова висока грануларност в сравнение с миналия регион, за да се стигне до задоволим резултат.

10000x10000, granularity test on default region, 8 tasks						
num threads	granularity	fastest thread	median	slowest	speedup	efficiency
1	1	39.639s	39.639s	39.639s	1	100%
8	1	3.631s	5.100s	6.828s	4.52	56.50%
8	2	4.940s	5.445s	5.656s	5.46	68.21%
8	4	4.017s	4.210s	4.215s	7.32	91.53%
8	8	3.994s	4.195s	4.124s	7.48	93.55%
8	16	3.994s	4.195s	4.198s	7.35	91.90%



Run-ване на програмата

```
$ git clone https://github.com/alexvitkov/mandelbrot/
$ cd mandelbrot
$ docker build -t mandelbrot .

$ docker run mandelbrot --size 2000x2000 --tasks 2 --granularity 4
$ docker cp /usr/src/mandelbrot/output.png .
```