

# CALCULANDO A ÁREA DO CONJUNTO DE MANDELBROT EM PARALELO COM OPENMP<sup>1</sup>

Lorenzo Manica<sup>2</sup> <lorenzo.manica@edu.pucrs.br>

Pontifícia Universidade Católica do Rio Grande do Sul – Faculdade de Informática – Curso de Ciência da Computação

Av. Ipiranga, 6681 Prédio 32 Sala 505 – Bairro Partenon – CEP 90619-900 – Porto Alegre – RS

19 de maio de 2022

## RESUMO

O presente artigo apresenta uma estratégia de paralelização do cálculo da área do Conjunto de Mandelbrot, fatiando segmentos do plano complexo e distribuindo em *threads* o processamento dos pontos, bem como uma forma de otimizar o processamento para distribuir melhor a carga de trabalho entre as *threads*. O estudo compara a performance da versão paralelizada não balanceada com uma versão efetuando o balanceamento. Os resultados mostram uma melhora no desempenho do cálculo em relação à versão sequencial do algoritmo e o aumento de eficiência obtido com o balanceamento da carga de trabalho.

Palavras-chave: Programação paralela, OpenMP, Mandelbrot.

## 1 INTRODUÇÃO

O Conjunto de Mandelbrot é um tópico bastante estudado na matemática, principalmente no campo da geometria fractal, que possui relação com diversas áreas do conhecimento, num espectro que vai desde as artes até a biologia. A relação com a área da Informática é particularmente especial, pois foi através do uso da computação que tornou-se possível o estudo mais aprofundado destas formas.

Os aspectos geométricos dos fractais apresentam características instigantes e possuem diversas aplicações, além de frequentemente despertarem a curiosidade das pessoas. Desta maneira, o estudo das suas propriedades geométricas torna-se interessante, pois, sendo o cálculo da área uma das propriedades mais elementares e, dado a sua estrutura complexa, apresenta um bom desafio para a computação e para a disciplina de processamento paralelo.

## 2 CÁLCULO DA ÁREA

A área do fractal de Mandelbrot pode ser medida por meio de uma discretização aproximada dos pontos do conjunto, de maneira que a sua precisão pode ser parametrizada arbitrariamente, dependendo do nível exigido pela aplicação. Obtém-se a discretização do conjunto através da contabilização dos pontos que não pertencem a ele, por meio de uma expansão da sequência que define o conjunto para cada ponto a fim de identificar quais deles excedem o valor de fuga no limite parametrizado pelo número de iterações. Para selecionar os pontos incluídos no cálculo, efetua-se uma varredura nos pontos contidos na região onde incide o conjunto, no intervalo  $[-2; 0,5]$  dos reais, com um raio  $r = 2$  a partir da origem incorporando o espaço do intervalo  $[-2, 2]$  dos imaginários no plano complexo. O cálculo da área é realizado subtraindo-se a área do retângulo na região de interesse no plano com o complemento da área formada pela exclusão dos pontos não pertencentes ao conjunto.

## 3 VERSÃO PARALELA

A primeira estratégia de paralelização consistiu em segmentar o processamento do laço mais externo do código, que itera sobre o eixo das abscissas, dividindo a região do plano em fatias verticais (Fig. 1). Esta característica foi obtida utilizando a diretiva *parallel for* do OpenMP.

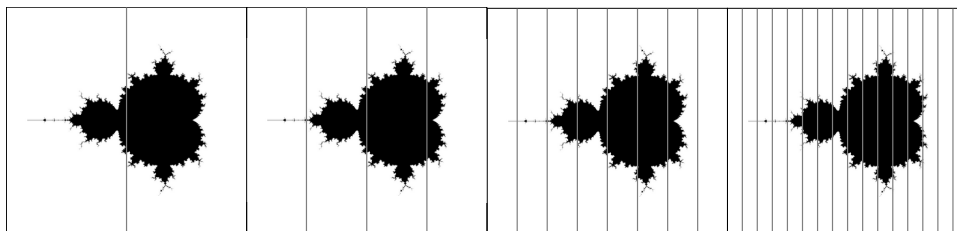


Figura 1. Segmentação do cálculo em regiões do eixo dos reais.

<sup>1</sup> Artigo elaborado como relatório da atividade de avaliação da disciplina Programação Paralela do curso de Ciência da Computação da Faculdade de Informática da PUCRS.

<sup>2</sup> Aluno da disciplina de Programação Paralela ministrada pelo prof. Roland Teodorowitsch.

Alguns cuidados tiveram de ser tomados no compartilhamento de dados a fim de manter a corretude do cálculo. Com a segmentação do trabalho, as variáveis das coordenadas dos pontos e a de contabilização da área de exclusão tiveram de ser replicadas e tornadas privadas dentro de cada tarefa. Outra medida de fundamental importância para a correta execução do cálculo foi a configuração da instrução de contabilidade como uma operação atômica, usando a diretiva *atomic*, no intuito de não permitir a concorrência de dados no referido trecho de código.

## 4 VERSÃO PARALELA BALANCEADA

Ao realizarmos uma rápida análise sobre este algoritmo, pode-se observar que o esforço de processamento do cálculo varia de acordo com as coordenadas do ponto, levando menos tempo para obter uma resposta nas regiões mais periféricas do plano, uma vez que os valores absolutos nessas regiões são mais altos e mais prováveis de exceder a condição de fuga do conjunto mais facilmente, evitando iterações adicionais nesse ponto. Em contrapartida, a região do conjunto de Mandelbrot é onde o processamento torna-se mais demorado, já que a propriedade do conjunto só é verificada após a execução de toda expansão da sequência, exaustivamente.

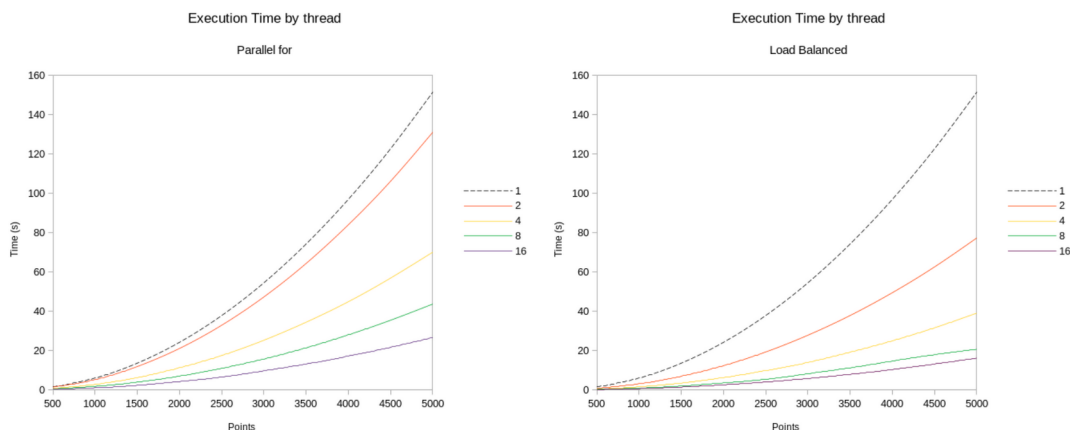
Em consequência disso, a carga de trabalho das tarefas de cálculo paralelizadas com a diretiva *parallel for* não é homogênea, fazendo com que o tempo de execução do algoritmo fique condicionado ao trabalho da tarefa mais carregada.

A fim de otimizar esse cenário, foi elaborada uma versão alternativa utilizando a cláusula *schedule* com as regras *monotonic*, *dynamic* em conjunto com a diretiva *parallel for*. Essa cláusula altera a forma como o trabalho é atribuído às tarefas, fazendo com que cada tarefa receba mais blocos de iterações a medida que as *threads* vão tornando-se ociosas.

## 5 ANÁLISE

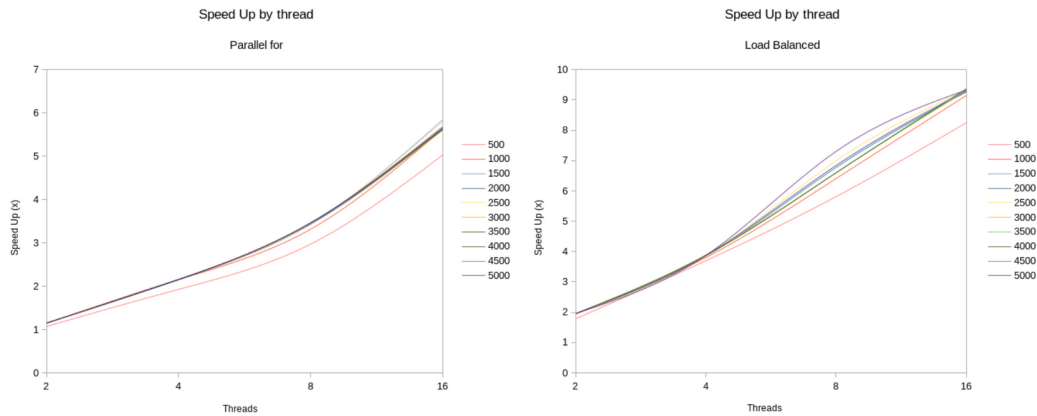
A fim de avaliar a performance do paralelismo nesta solução, foram elaborados diversos testes com configurações variando em grau de paralelismo e em resolução do conjunto. Foram testados quatro graus de paralelismo (2, 4, 8 e 16) e quantidades de pontos variando entre 500 e 5000, em incrementos de 500 pontos.

Primeiramente foi efetuada a execução da versão sequencial do algoritmo para que fosse possível ter uma base de comparação entre as versões. A versão sequencial levou 1,52s para 500 pontos e 151,53s para calcular 5000 pontos (Tabela 1). Nas versões paralelas, foram realizadas 5 execuções para cada cenário descrito, obtendo a média aritmética das execuções. Todos os testes foram executados em processadores Intel Xeon com 16 núcleos. A versão paralela obteve melhora nos tempos de execução em todos os cenários, em especial no caso com grau de paralelismo 16, levando 26,68s para calcular 5000 pontos. A versão paralela balanceada obteve um desempenho ainda melhor, levando 16,24s para calcular 5000 pontos em grau de paralelismo 16, conforme a Figura 2.

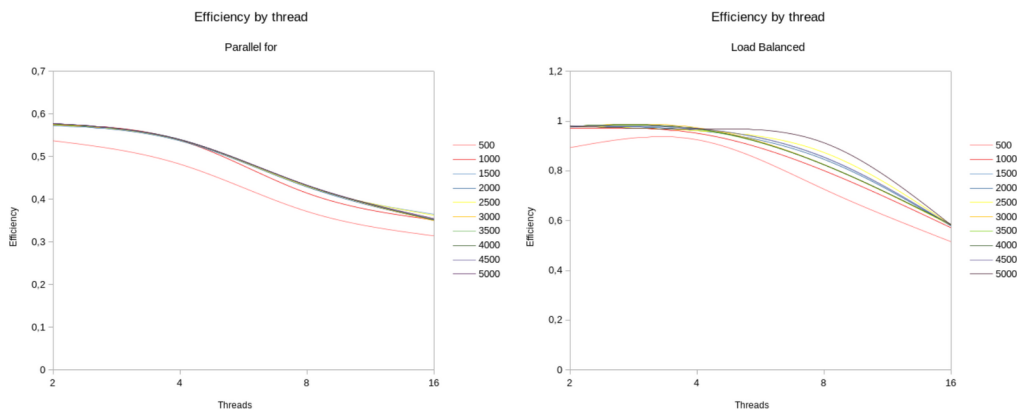


**Figura 2. Tempos de execução versus número de threads das versões paralela e paralela balanceada.**

Ao calcular-se o fator de aceleração obtido pela versão paralela na execução dos cenários previstos, percebeu-se um leve aumento na aceleração, na ordem de 1,15x da versão sequencial com grau de paralelismo 2, aumentando progressivamente para 2,16x, 3,46x e 5,67x com os graus de paralelismo 4, 8, e 16, respectivamente. Já com a versão balanceada foi possível observar um aumento mais expressivo e linear no fator de aceleração, atingindo um aumento na ordem de 1,95x com grau 2, 3,87x (grau 4), 7,30x (grau 8) e finalmente, 9,32x com grau de paralelismo 16 (Fig. 3).



**Figura 3. Speed up versus threads das versões paralela e paralela balanceada**



**Figura 4. Eficiência versus número de threads das versões paralela e paralela balanceada**

Os níveis de *Speed up* obtidos na versão paralela não balanceada são justificados pela baixa eficiência da distribuição da carga de trabalho estática da diretiva *parallel for*. O gráfico de eficiência (Fig. 4) aponta um aproveitamento do tempo de processador na casa de 57%, regredindo para 54%, 43% e 35% nessa versão para os graus de paralelismo 2, 4, 8 e 16, respectivamente. Na versão com balanceamento dinâmico de carga, por esta utilizar a ociosidade dos núcleos como regra de distribuição de trabalho, e consequentemente conseguir ocupar melhor o tempo de processador durante o cálculo, os resultados são expressivamente melhores, obtendo índices de eficiência próximas do ideal para graus 2, 4 e 8 de paralelismo (97%, 96% e 91%, respectivamente), e posteriormente caindo para um nível de 58% com grau de paralelismo 16.

	REF.	PARALELO				BALANCEADO			
p		2	4	8	16	2	4	8	16
500	1,52278388	1,416883374	0,789139686	0,512250234	0,302852204	0,852251236	0,411407874	0,262011816	0,18462424
1000	6,07285026	5,280967986	2,81819122	1,830732854	1,0794851	3,130436592	1,595817792	0,947595264	0,664067372
1500	13,64983285	11,929119534	6,312927682	3,973894616	2,337619768	6,994406834	3,521499074	1,998059706	1,472042428
2000	24,25913699	21,141269938	11,29170896	7,065265674	4,319806012	12,393355514	6,302114188	3,582034932	2,61888251
2500	37,89312889	32,959903668	17,550795602	10,995893852	6,534179342	19,338135208	9,849277702	5,418526732	4,061819438
3000	54,56576432	47,359255888	25,312655678	15,77595941	9,751996468	27,870256478	14,012522794	8,26499701	5,86181353
3500	74,27799999	64,380678044	34,476143178	21,45348117	13,192080146	37,890485994	19,20150589	11,25089954	7,977245818
4000	97,0034656	84,054416822	44,899542898	28,047860368	17,222062526	49,492469942	25,007519764	14,7033056	10,358810878
4500	122,75262951	106,202283016	56,918018102	35,474484958	21,711121144	62,61534315	31,631148236	17,93167717	13,193745442
5000	151,53928168	131,160034366	70,118870426	43,714575634	26,68274119	77,349928262	39,110963492	20,746199054	16,244905742

**Tabela 1. Tempos de Execução**

## 6 CONCLUSÃO

O balanceamento de carga dinâmico do OpenMP usando a diretiva *schedule (monotonic, dynamic)* mostrou-se uma boa opção para otimizar o cálculo da área do Conjunto de Mandelbrot, aumentando significativamente a performance do algoritmo e atingindo níveis de eficiência muito bons. Contudo, outras diretivas do OpenMP oferecem potencial para melhorar ainda mais a performance do cálculo, em particular a estratégia de redução com diretiva *reduce(sum)*, visto que a integração da área passa por efetuar o somatório das parcelas da área delimitada pela curva do fractal, embora essa mudança de abordagem exija modificações no código e possivelmente na maneira de estruturar o algoritmo.

## REFERÊNCIAS

OPENMP. OpenMP Reference Guide. Disponível em <<https://www.openmp.org/wp-content/uploads/OpenMPRefCard-5-2-web.pdf>>. Acesso em 19 mai. 2022

WOLFRAM ALPHA. Mandelbrot Set. Disponível em: <<https://www.wolframalpha.com/input?i=Mandelbrot+set>>. Acesso em: 19 mai. 2022.

WIKIPEDIA. Mandelbrot Set. Disponível em: <[https://en.wikipedia.org/wiki/Mandelbrot\\_set](https://en.wikipedia.org/wiki/Mandelbrot_set)>. Acesso em: 19 mai. 2022.