

Estatísticas de Desempenho na Geração de Fractais em Máquinas Multiprocessadas*

Diego Luís Kreutz, Fábio Natanael Kepler, Benhur Stein

Universidade Federal de Santa Maria – UFSM

Laboratório de Sistemas de Computação

{kreutz,kepler,benhur}@inf.ufsm.br

Introdução

A cada dia surgem mais aplicações que necessitam de um alto poder de processamento. Com isto emerge a exigência de ferramentas que auxiliem a implementação de algoritmos mais eficientes. Dentro deste contexto entra a análise de estatísticas do comportamento que a aplicação apresentou segundo parâmetros controlados pelo usuário. Neste trabalho buscou-se desenvolver uma ferramenta para o cálculo de fractais que fornece um meio de o usuário visualizar o comportamento da aplicação segundo a manipulação dinâmica dos parâmetros de cálculo e divisão das tarefas. Esta ferramenta tem por objetivo principal a visualização de estatísticas de cálculo de fractais em máquinas multiprocessadas a fim de fornecer dados para o desenvolvimento de um algoritmo mais eficiente. Esta ferramenta introduz a possibilidade de identificação de regiões que demandam um maior tempo de processamento, por exemplo. A possibilidade de identificação destas regiões torna factível ao usuário o desenvolvimento de uma aplicação mais eficiente para cálculo de fractais.

A fim de obter uma visualização mais detalhada e fornecer estatísticas de desempenho foram implementadas janelas gráficas contendo dados estatísticos do estado da execução do programa. Com estes dados estatísticos é possível também realizar cálculos de sobrecarga paralela do sistema e gerar comparações de desempenho em relação a algoritmos sequenciais e a diferentes parametrizações aplicadas ao algoritmo concorrente.

As seções seguintes abrangem uma breve descrição dos fractais e uma descrição mais detalhada da aplicação.

Fractais

Fractais são formas geométricas com alguma semelhança entre si, mas de estruturas muito peculiares. Um aspecto importante de um fractal é que ele possui uma dimensão mínima. Ao invés de uma, duas ou três dimensões (como é o caso dos objetos mais familiares), a maioria dos fractais possui dimensões que não se enquadram nessa concepção. Todavia, sua dimensão raramente pode ser expressada como um valor inteiro. Isto é precisamente o que esta por trás do seu nome.

As imagens de fractais são tão coloridas e bizarras devido a atribuição de uma faixa de cores possíveis a uma série de pontos dependendo do comportamento destes pontos segundo a resolução da função que os expressa, com o auxílio de um computador.

*Trabalho desenvolvido na cadeira de Sistemas Operacionais.

Um exemplo de uma possível aplicação do cálculo de fractais é a mensuração de fatores associados ao crescimento desordenado de um tumor cancerígeno. Este cálculo poderia auxiliar o tratamento e diagnóstico do tumor. Outro exemplo é a aplicação da geometria de fractais a problemas da indústria de petróleo. Esta utiliza a geometria de fractais em algumas técnicas de captação de dados, interpolação de dados referentes a abismos geográficos e dados sísmicos, avaliação de bacias de captação e formas de acesso.

Hoje, graças à descoberta da teoria do caos e a geometria dos fractais, cientistas estão capacitados a entender como sistemas totalmente caóticos possuem formas previsíveis, por exemplo.

O Gerador de Fractais

A idéia básica da implementação desta aplicação consiste na divisão da tarefa de calcular o fractal em subtarefas. Esta aplicação possui diversos parâmetros controláveis dinamicamente para facilitar a realização dos ajustes necessários para que se atinja o máximo desempenho na execução do algoritmo. O controle dinâmico permite também a visualização dos efeitos causados sobre a execução do programa a partir da alteração de algum destes parâmetros (veja maiores detalhes e quais são os parâmetros controláveis na seção).

As seções que seguem descrevem a forma de divisão das tarefas e a visualização dos dados estatísticos gerados.

Divisão e distribuição de tarefas

A tarefa principal a ser realizada é o cálculo de um fractal. Para otimizar este cálculo divide-se esta tarefa em tarefas menores utilizando o modelo mestre-escravo. Este modelo consiste em um fluxo de execução principal (*thread* pai) que controla a divisão e produção das tarefas bem como a criação dos fluxos de execução secundários (*threads* filhos) que serão responsáveis pela execução das tarefas e geração dos resultados. O *thread* pai é responsável pela divisão da tarefa principal em tarefas menores, de acordo com parametrizações do usuário, a serem realizadas pelos *threads* filhos. Cada tarefa gerada pelo *thread* pai representa uma determinada região do fractal.

Os *threads* filhos disputam as tarefas a serem realizadas. A partir da obtenção de uma tarefa a *thread* filho passa a executá-la. Após a finalização do cálculo, retorna o resultado e volta a disputar uma nova região até que todas as tarefas tenham sido executadas.

Visualização de Estatísticas

Paralelo à distribuição, cálculo e exibição do fractal é feito o cálculo do tempo que cada *thread* filho gasta calculando regiões, o número de regiões e quais as regiões que cada processo calculou. Essas informações são exibidas na janela de visualização (ver figura 1), além do fractal, os gráficos do tempo de processador, do número de regiões e as regiões calculadas por cada *thread* filho.

Na figura 1, a tela superior direita representa uma visualização do fractal de Mandelbrot. A tela superior esquerda representa as *threads*, onde cada *thread* é representada por uma cor, e as regiões que cada uma calculou. A tela inferior esquerda apresenta o

número de regiões que cada *thread* calculou. Por fim, a tela inferior direita apresenta o gráfico do tempo de processamento de cada *thread*.

Análise de desempenho

Aplicações seqüenciais, como por exemplo o cálculo de fractais, normalmente necessitam de um enorme poder computacional para serem realmente factíveis. Com a paralelização destas aplicações atinge-se um nível de desempenho bastante superior ao modo convencional. Por exemplo, um programa para cálculo de fractais que utilize uma faixa de 4 bilhões de iterações para a determinação da cor de cada pixel levaria duas semanas para completar o cálculo. Por outro lado, um programa idêntico, porém paralelo, que utilize o poder computacional de 4 máquinas biprocessadas levaria, proporcionalmente, mais o menos 3 dias e meio. Portanto, esta simples ilustração demonstra que a paralelização de aplicações pode ser muito útil na elaboração de programas que necessitam de um alto poder computacional.

Através da comparação da execução do programa com parâmetros diferentes pode-se observar a eficiência da aplicação. O número de *threads*, o número de regiões (subdivisão), o tamanho das regiões e a profundidade de cálculo são os parâmetros que podem ser alterados dinamicamente pelo usuário.

Um número grande de *threads* ou regiões muito pequenas geram uma grande sobrecarga no sistema. Já um número pequeno de *threads* ou regiões muito grandes resultam em um tempo de processamento, por tarefa, bastante elevado, desperdiçando capacidade de processamento de máquinas multiprocessadas, ou seja, ocorre um desequilíbrio na execução da aplicação. Cabe ao usuário determinar o número ideal de *threads*, o tamanho ideal das regiões e o número ideal de regiões.

O parâmetro da profundidade permite ao usuário determinar o nível de detalhes a ser calculado. Muitas vezes, com um nível de detalhes muito baixo, não consegue-

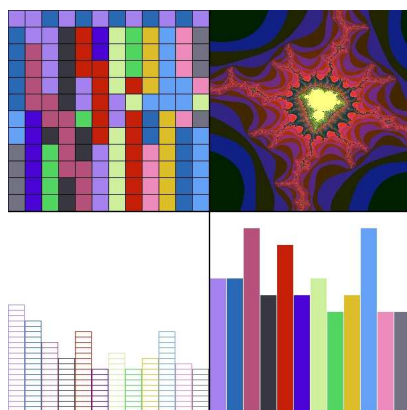


Figura 1: janela de visualização da aplicação.

se identificar as regiões mais custosas de cálculo. Com a possibilidade de se identificar estas regiões mais custosas existe a oportunidade do desenvolvimento de aplicações mais eficientes através da divisão destas regiões em tarefas ainda menores a fim de utilizar melhor o potencial computacional disponível em máquinas multiprocessadas.

As informações que podem ser disponibilizadas pela aplicação durante e após o cálculo e exibição de um fractal são o número de *threads* filhos disparados, o tempo de processador de cada um deles, as regiões que cada um calculou e o número de regiões calculadas, o tamanho das regiões e a profundidade do cálculo.

De posse dessas informações, e analisando a evolução do cálculo do fractal, poderíamos derivar facilmente outras informações como o sobrecarga paralela do sistema.

Conclusão

Este programa disponibiliza uma forma fácil e intuitiva de identificar as regiões que demandam maior custo de processamento, de determinar o número ótimo de processos e de mensurar o tamanho ideal das regiões através de gráficos de estatísticas de execução.

Com a identificação das regiões mais custosas é possível melhorar substancialmente o desempenho do algoritmo.

O desenvolvimento de aplicações mais específicas, como o cálculo do comportamento de determinados tumores cancerígenos, pode agilizar o processo de diagnosticação e cura, por exemplo. A aplicação desenvolvida neste trabalho auxilia o desenvolvimento de aplicações deste gênero.

A incorporação de funções destinadas a geração de rastros de execução, a fim de visualizar, utilizando uma ferramenta adequada, o comportamento dinâmico que o programa obteve durante sua execução, e a implementação da ferramenta utilizando um sistema de memória distribuída são algumas das tarefas previstas como trabalhos futuros.

Referências

- [BOD 2000] BODE, A. et al. **Euro-Par 2000 Parallel Processing**. Munich, Germany: 6th International Euro-Par Conference, 2000.
- [CHA 2000] CHAFFEY High School's FRACTALS on the Web. <http://www.chaffey.org/fractals/index.html>.
- [CHA 2000a] CHAOSPRO. www.chaospro.de.
- [DIV 2001] DIVERIO, T. A.; NAVAUX, P. O. A. **ERAD2001 - Primeira Escola Regional de Alto Desempenho**. Anais.
- [JON 2000] JONES, D. M. **Fractalus**. www.fractalus.com.
- [OLI 2000] OLIVEIRA, R. S. de; SILVA CARISSIMI, A. da; TOSCANI, S. S. **Sistemas Operacionais**. [S.l.]: Instituto de Informática da UFRGS, 2000.
- [TAN 97] TANENBAUM, A. S.; WOODHULL, A. S. **Operating Systems Design and Implementation**. 2.ed. [S.l.]: Prentice Hall, 1997.