Universidade Federal de Ouro Preto - UFOP

Instituto de Ciências Exatas e Biológicas - ICEB

Departamento de Computação - DECOM

USO DE PARALELISMO DE DADOS PARA MAIOR EFICIÊNCIA DE ALGORITMOS DE PROCESSAMENTO DE IMAGENS

Aluno: Pedro Ribeiro Mendes Júnior
Matrícula: 08.2.4114

Orientador: Lucília Camarão de Figueiredo

Ouro Preto
26 de setembro de 2011

Universidade Federal de Ouro Preto - UFOP

Instituto de Ciências Exatas e Biológicas - ICEB

Departamento de Computação - DECOM

USO DE PARALELISMO DE DADOS PARA MAIOR EFICIÊNCIA DE ALGORITMOS DE PROCESSAMENTO DE IMAGENS

Proposta de monografia apresentada ao curso de Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, como requisito parcial para a conclusão da disciplina Monografia I (BCC390).

Aluno: Pedro Ribeiro Mendes Júnior
Matricula: 08.2.4114

Orientador: Lucília Camarão de Figueiredo

Ouro Preto
26 de setembro de 2011

Resumo

O projeto busca explorar oportunidades de uso de paralelismo de dados em algoritmos voltados para processamento de imagens, com o objetivo de obter algoritmos mais eficientes para essas aplicações. Para isso, propomos desenvolver algoritmos de processamento de imagem usando a linguagem funcional Haskell, em particular Data Parallel Haskell, uma extensão do compilador GHC e de suas bibliotecas, que oferece suporte para paralelismo aninhado com foco na utilização de CPUs multicore. Os dados de tempo de execução dos algoritmos desenvolvidos serão medidos e comparados com o de outras implementações existentes. No caso dessas comparações indicarem bons resultados, planeja-se, como trabalho futuro, a implementação de uma biblioteca de funções para processamento de imagens com enfoque em paralelismo, a qual deverá ser útil para implementação eficiente de várias aplicações na área.

Palavras-chave: Paralelismo de dados. Processamento digital de imagens. Data Parallel Haskell.

Sumário

Lista de Tabelas

1 Introdução

Este trabalho visa estudar oportunidades de explorar a capacidade de processamento paralelo dos computadores modernos, com o objetivo de obter algoritmos mais eficientes para aplicações de processamento de imagens.

A arquitetura dos computadores modernos, inclusive PCs, é baseada em múltiplos processadores, sendo um exemplo particularmente importante o de unidades de processamento gráfico (GPUs) [6], constituídas de dezenas de processadores, e altamente eficientes para a manipulação de imagens gráficas. A estrutura altamente paralela de GPUs as torna também mais efetivas do que as CPUs de propósito geral para algoritmos nos quais pode ser explorado o processamento de grandes blocos de dados em paralelo.

Várias formas de paralelismo podem ser exploradas na execução de programas. O mais comum é o paralelismo de controle, no qual se criam explicitamente *threads* que executam paralelamente e cujo processamento é sincronizado por meio de mensagens, *locks* ou outros mecanismos de sincronização [8]. O paralelismo de dados consiste na execução simultânea de uma mesma instrução em diferentes porções de um conjunto de dados, sendo esta abordagem particularmente promissora para o emprego de grande número de processadores [2].

O trabalho pioneiro de Blelloch em NESL [1] mostrou que é possível combinar um modelo de programação bastante flexível - *nested data paralelism* - com um modelo de execução eficiente e de grande escalabilidade - *flat data paralelism* -, ampliando o leque de possibilidades para explorar paralelismo em programas. Essa abordagem é implementada em Data Parallel Haskell [3], o *framework* de programação que propomos utilizar para o desenvolvimento deste projeto, e que consiste em uma extensão do compilador GHC [4] da linguagem Haskell [5] e de suas bibliotecas, de maneira e embutir na linguagem paralelismo de dados aninhado.

Uma área na qual parece ser particularmente fértil explorar essa forma de paralelismo para obter implementações mais eficientes é a área de processamento de imagens. Alguns exemplos de sucesso na implementação de algoritmos paralelos para processamento de imagens são reportados, por exemplo, em [7, 9].

2 Justificativa

O projeto pretende apresentar contribuições relativas a uma questão extremante atual e importante, que consiste em estudar, desenvolver e avaliar técnicas para explorar a capacidade de processamento dos computadores modernos, com arquitetura baseada em múltiplos processadores, no sentido de obter algoritmos e implementações mais eficientes de programas.

3 Objetivos

3.1 Objetivo geral

O objetivo geral do projeto é explorar oportunidades de uso de paralelismo de dados aninhados para a implementação eficientes de algoritmos.

3.2 Objetivos específicos

- Identificar algoritmos de processamento de imagem para os quais possam ser obtidas implementações mais eficientes, por meio do uso de paralelismo.
- Identificar o tipo de paralelismo mais adequado em cada caso.
- Obter implementações paralelas eficientes desses algoritmos, que possam ser usadas em aplicações reais.
- Obter evidências a favor ou contra o argumento de que linguagens funcionais são particularmente apropriadas para a implementação de paralelismo.

4 Metodologia

1. Estudos dos diversos modelos de paralelismo, incluindo paralelismo de controle, paralelismo de dados, e paralelismo de dados aninhados.

2. Seleção de aplicações interessantes de processamento de imagens, que possam apresentar melhoria de eficiência se implementadas usando paralelismo.

3. Instalação e configuração de plataformas de desenvolvimento, em particular, Data Parallel Haskell.

4. Estudo da linguagem e bibliotecas de Data Parallel Haskell.

5. Implementação de alguns algoritmos selecionados no item 2.

6. Comparação entre a implementação desenvolvida neste projeto e implementações que não exploram paralelismo, e também com outras implementações pararelas eventualmente existentes.

7. Avaliação dos resultados obtidos, em termos de eficiência, tipo de paralelismo explorado, novas oportunidades de paralelismo, etc.

8. Redação de relatório técnico/artigo descrevendo o trabalho desenvolvido e seus resultados.

5 Cronograma de atividades

As atividades estão planejadas para serem realizadas de acordo com a Tabela 1.

Atividades	**Ago**	**Set**	**Out**	**Nov**	**Dez**
Estudo dos diversos modelos de paralelismo		X	X		
Seleção de aplicações	X	X	X	X	
Estudo das bibliotecas	X	X	X		
Implemetação de algoritmos	X	X	X	X	
Comparações entre implementações				X	X
Avaliação dos resultados				X	X
Redigir a monografia				X	X
Apresentação do trabalho					X

Tabela 1: Cronograma de Atividades.

Referências

[1] Guy E. Blelloch. Programming parallel algorithms. *Communications of the ACM*, 39(3):85–97, 1996.

[2] Manuel M. T. Chakravarty, Gabriele Keller, Sean Lee, Trevor L. McDonell, and Vinod Grover. Accelerating haskell array codes with multicore gpus. In *Proc. of Declarative Aspects of Multicore Programming*, 2011.

[3] Manuel M. T. Chakravarty, Roman Leshchinskiy, Simon Peyton Jones, Gabriele Keller, and Simon Marlow. Data parallel haskell: a status report. In *Proc. of ACM Workshop on Declarative Aspects of Multicore Programming*, Nice, 2007.

[4] The Glasgow Haskell Compiler. `http://haskell.org/haskellwiki/GHC`.

[5] Simon Peyton Jones. Haskell report. Technical report, Cambridge University Press, 2003.

[6] Rainer Keller, David Kramer, and Jan-Philipp Weiss, editors. *Facing the Multicore-Challenge.* Aspects of New Paradigms and Technologies in Parallel Computing Series. Lecture Notes in Computer Science, first edition, 2010.

[7] Ben Lippmeier, Gabriele Keller, and Simon Peyton Jones. Efficient parallel stencil convolution in haskell. In *Proc. of International Conference on Functional Programming*, 2011.

[8] Peter Pacheco. *An Introduction to Parallel Programming.* Morgan Kaufmann, first edition, 2011.

[9] F. J. Seinstra and D. C. Koelma. User transparency: A fully sequential programming model for efficient data parallel image processing. *Concurrency and Computation: Practice and Experience*, 16(6):611–644, 2004.