

Artigo pro Cuda Fuzzy

Edevaldo Braga dos Santos¹, Giovane de Oliveira Torres¹, Guilherme Pereira Paim¹,
Renan Zafalon da Silva¹, Vitor Alano de Ataides¹, Maurício Lima Pilla¹

¹Universidade Federal de Pelotas
Pelotas, RS - Brasil

{edevaldo.santos,gdotorres,gppaim,renan.zafalon,vaataides,pilla}@inf.ufpel.edu.br

Abstract. *Abstract aqui.*

Resumo. *Resumo aqui.*

1. Introdução

Existem diversos casos onde classes de objetos não pertencem totalmente a um conjunto. Baseado nisto, Zadeh definiu a teoria dos conjuntos *fuzzy* [Zadeh 1965], o que visa tratar problemas de imprecisão ao classificar dados no mundo real. Os conjuntos *fuzzy* possuem aplicações em sistemas de controle e de suporte à decisão, onde a descrição do problema não é feita de forma precisa [Weber and Klein 2003].

Utilizando-se dos conjuntos *fuzzy*, tem-se a base para a lógica *fuzzy*, sendo construído a partir da lógica proposicional. Com isto, os operadores foram definidos a partir dos já estabelecidos na lógica clássica, com a adição de outros para fins práticos [Tanscheit 2004]. Uma característica interessante que diferencia a lógica tradicional da *fuzzy* é que na primeira os valores que são utilizados atendem a condição de serem verdadeiros ou falsos (0 ou 1). Já na segunda, trabalha-se com conjuntos *fuzzy* – estes podem assumir um valor que pertence ao intervalo $[0, 1]$, o que permite que um conjunto *fuzzy* possa ser representado por uma infinidade de valores [Klir and Yuan 1995].

A fim de obter-se computação com bom desempenho, é importante fazer uso dos vários núcleos de processamento os quais são disponibilizados nos sistemas de computação atuais – para poder por em prática o uso do paralelismo. Neste contexto, encaixam-se as GPUs (*Graphical Processor Units*), as quais são componentes com alto poder de paralelismo [Sengupta et al. 2007]. Porém, é importante ressaltar que as GPUs são reservadas a obter bom desempenho com aplicações que possuem determinadas características [Owens et al. 2008] que incluem: (i) Requisitos computacionais grandes, (ii) Paralelismo nas aplicações e (iii) maior importância ao *throughput* do que a latência. Destacam-se alguns exemplos práticos bem-sucedidos que utilizam CUDA: Análise do fluxo de tráfego aéreo, através do uso do poder computacional de CUDA, foi possível reduzir o tempo de análise do tráfego aéreo nacional de dez minutos para três segundos. Outro exemplo relevante é o ganho de performance em simulações moleculares NAMD (dinâmica molecular em nanoescala), o ganho de performance foi possível graças as arquiteturas paralelas das GPUs [NVIDIA 2015].

Tendo estes conceitos discutidos, o objetivo deste trabalho é descrever uma biblioteca de lógica *fuzzy* voltada para GPUs, a fim de verificar como pode ser efetuado uma implementação que consiga extrair paralelismo deste tipo de arquiteturas.

Referências

- Klir, G. and Yuan, B. (1995). *Fuzzy sets and fuzzy logic*, volume 4. Prentice Hall New Jersey.
- NVIDIA (2015). Site do nvidia. Disponível em: <<http://>>.
- Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). Gpu computing. *Proceedings of the IEEE*, 96(5):879–899.
- Sengupta, S., Harris, M., Zhang, Y., and Owens, J. D. (2007). Scan primitives for gpu computing. In *Graphics hardware*, volume 2007, pages 97–106.
- Tanscheit, R. (2004). Sistemas fuzzy. *Inteligência computacional: aplicadaa administração, economia e engenharia em Matlab*, pages 229–264.
- Weber, L. and Klein, P. A. T. (2003). *Aplicação da lógica fuzzy em software e hardware*. Editora da ULBRA.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.