

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* [?], 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 [?].

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 [?]. 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 [?].

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 [?]. Porém, é importante ressaltar que as GPUs são reservadas a obter bom desempenho com aplicações que possuem determinadas características [?] 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.

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.