

# Artigo pro Cuda Fuzzy

Edevaldo Braga dos Santos<sup>1</sup>, Giovane de Oliveira Torres<sup>1</sup>, Guilherme Pereira Paim<sup>1</sup>,  
Renan Zafalon da Silva<sup>1</sup>, Vitor Alano de Ataides<sup>1</sup>, Maurício Lima Pilla<sup>1</sup>

<sup>1</sup>Universidade 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 desempenho em simulações moleculares NAMD (dinâmica molecular em nanoescala), o ganho de desempenho 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.

O restante deste artigo está dividido da seguinte maneira: A seção 2 fala sobre a lógica fuzzy, que é a base para a construção deste trabalho. Na seção 3, é descrito a implementação efetuada da biblioteca CudaFuzzy. A seção 4 destina-se a explicar a metodologia empregada para a execução de testes na biblioteca. A seção 5 exhibe e discute os principais resultados obtidos por este trabalho, de onde se tiram as principais conclusões, observadas na seção 6, a qual ainda mostra possíveis trabalhos futuros. Por fim, a seção 7 faz uma breve discussão sobre os trabalhos relacionados ao escopo deste artigo.

## **2. Lógica Fuzzy**

Lógica fuzzy foi criada com base na teoria de conjuntos Fuzzy, a ideia principal de fuzzy é a seguinte: Um valor é verdadeiro (grau 1) ou falsos (grau 0), sendo que o grau pode variar de 0 à 1. A teoria do conjunto fuzzy foi inventada com o objetivo de oferecer ferramentas matemáticas para solucionar problemas imprecisos ou vagos. Dentre as inúmeras aplicações com lógica fuzzy, o projeto IMMO-RATE, é bastante interessante, pois ele permite análise de sustentabilidade em imóveis, considerando questões chaves específicas que utilizam a lógica fuzzy.

### **2.1. Operadores Fuzzy**

Conjuntos são definidos por uma condição específica que define se um conjunto pertence ou não a um conjunto. Para exemplificar os operadores em fuzzy logo abaixo foram utilizados conjuntos A e B. Os operadores mais comuns em conjuntos fuzzy são: união ( $A \cup B = x \mid x \in A \vee x \in B$ ) e intersecção ( $A \cap B = x \mid x \in A \wedge x \in B$ ).

## **3. CudaFuzzy**

## **4. Metodologia**

## **5. Resultados e Discussão**

## **6. Conclusões e Trabalhos Futuros**

## **7. Trabalhos Relacionados**

Existem diversas implementações relacionadas à lógica *fuzzy*. Os artigos visualizados na bibliografia normalmente fazem utilização de lógica *fuzzy* voltada para um tipo de problema, não descrevendo uma biblioteca genérica. Como trabalhos descritos desta forma, existe [?], utiliza-se lógica *fuzzy* para a discussão de um método para modelagem qualitativa. Em [?], é empregada a lógica *fuzzy* para fazer o gerenciamento de energia e bateria de automóveis híbridos do tipo *plug-in*.

Como bibliotecas relacionadas à lógica *fuzzy*, existe uma desenvolvida na linguagem Java, chamada de jFuzzyLogic [?, ?]. Esta biblioteca é uma implementação de sistemas *fuzzy* que permite projetar controladores de lógica *fuzzy*. Por fim, existe a biblioteca FuzzyGPU [?], a qual é o trabalho relacionado mais próximo ao que este artigo apresenta. FuzzyGPU é uma implementação de biblioteca de aritmética *fuzzy* voltada à GPUs.