



Assignment 2 – Cyclic Cross Correlation

Presentation

Sebastian González - 103690

Bernardo Marçal - 103236

sebastian.duque8@ua.pt

bernardo.marcal@ua.pt

DETI

Departamento de Electrónica, Telecomunicações e Informática

26 de Dezembro de 2023

- Cyclic Cross Correlation é uma ferramenta importante para detectar semelhanças entre sinais discretos.
- A correlação circular é definida pela seguinte fórmula:

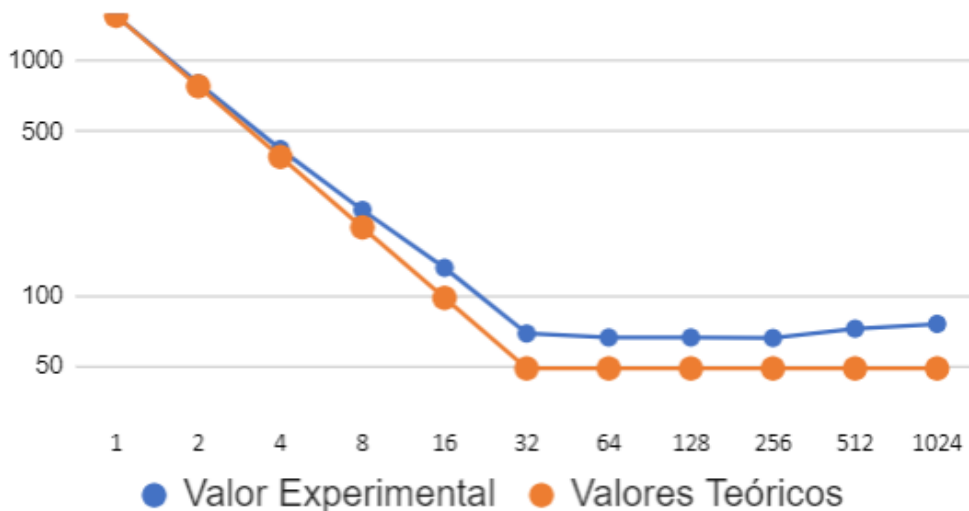
$$r_{xy}(\tau) = \sum_{k=0}^{n-1} x(k) \cdot y[(\tau + k) \text{ mód } n]$$

- O objetivo do Assignment 2 é otimizar o lançamento de threads e concluir sobre a utilidade de transferir o cálculo para um GPU comparando o com a performance de um CPU
- Para isso utilizamos a API CUDA que fornece uma interface de configuração de um processador gráfico

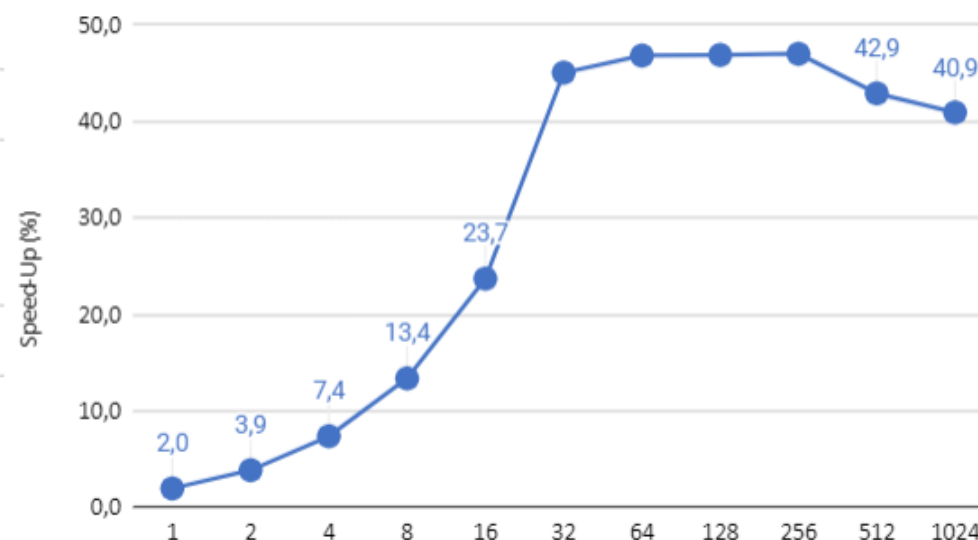
Variação blockDim e gridDim

- Para otimizar o processamento na GPU foi feita uma variação dos valores blockDim e gridDim no cyclicCircConv.cu para verificar em qual caso resulta numa maior performance.
- Avaliando os gráficos obtidos pode se concluir que valor da performance aumenta exponencialmente com o aumento dos threads até às 32. A partir deste valor temos uma atenuação do speedUp até às 1024.
- Também se concluiu que a performance atingiu o seu valor máximo com 256 threads.

Variação dos valores block/grid



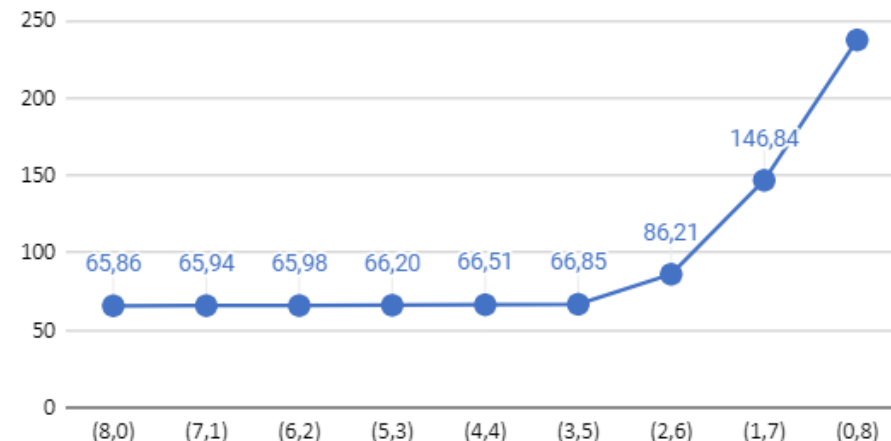
Speed-Up



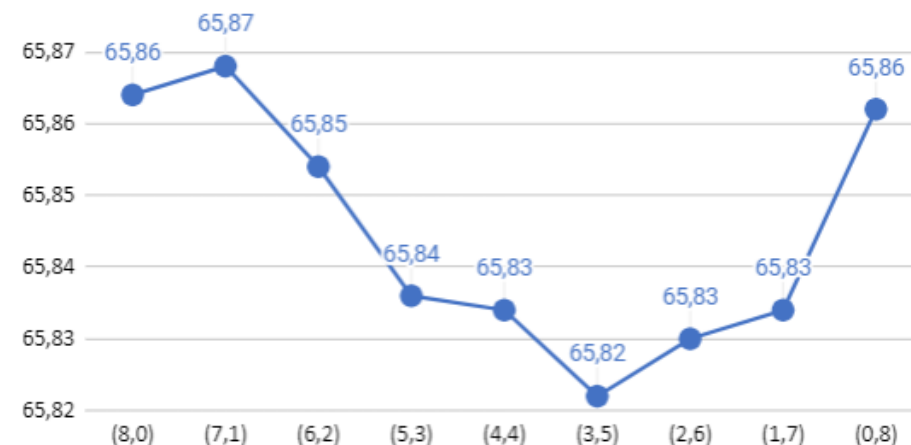
Variação blockDim e gridDim

- Como dito anteriormente, testámos todas as combinações dos valores do BlockDim para a melhoria da performance.
- Conclui-se que BlockDimX= 8 e BlockDimY = 0 com BlockDimZ default é o valor mais eficiente.
- Com os valores de blockDim constantes, foram variados os valores de gridDim e concluiu-se que a variação destes valores grande não teve impacto na performance.
- Apesar de não se notar um grande impacto, concluímos que a configuração mais eficiente é com o gridDimX = 3 e gridDimY = 5.

Variação dos valores de blockDim



Variação dos valores de GridDim



- Tempo de execução com acesso à GPU (**T GPU**):
gpuExecTime + cudaKernel + dataTransfer
- Tempo de execução com acesso à CPU (**T CPU**): cpuKernel

$$\text{SpeedUp} = T_{\text{GPU}} + T_{\text{CPU}}$$

$$= 3.120 / (0.0001224 + 0.1297 + 0.06582)$$

$$\approx 16 \text{ vezes mais rapido}$$

- Resumindo, constatamos que a execução de cálculos intensivos numa GPU é significativamente mais veloz e eficaz em comparação com a CPU, mesmo levando em conta os períodos de transferência de dados.