



# Trabalho 3 de CLE

Grupo 9 TP2

Diogo Andrade 89265

Francisco Silveira 84802



# Introdução

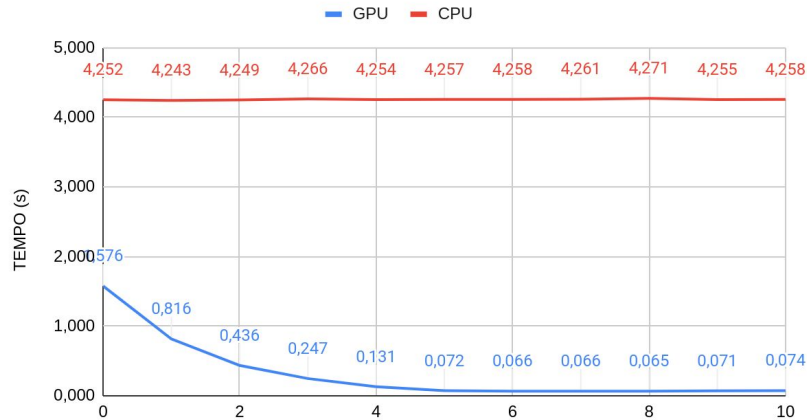
O objectivo deste trabalho é transformar as antigas implementações (multithread e multiprocessing) de cálculo da “circular cross-correlation” de cada par de sinais, numa implementação com CUDA.

Para isso, inicialmente são gerados dois arrays com valores do tipo *double* aleatórios. De seguida, tanto o GPU como o CPU irão calcular a “circular cross-correlation”.

Após feitos todos os cálculos, será comparado o resultado do GPU com o do CPU, caso os resultados obtidos não sejam iguais irá aparecer uma mensagem a informar.

# Exercício 1 - Resultados

MÉDIA DOS TEMPOS DE EXECUÇÃO



```
./cccCuda Starting...  
Using Device 0: GeForce GTX 1660 Ti  
Total samples data size: 524288  
The initialization of host data took 1.661e-03 seconds  
The transfer of 1572864 bytes from the host to the device took 2.026e-04 seconds  
The CUDA kernel <<<(256,1,1), (256,1,1)>>> took 6.542e-02 seconds to run  
The transfer of 524288 bytes from the device to the host took 2.311e-04 seconds  
The cpu kernel took 4.274e+00 seconds to run (single core)  
All is well!
```

Figura 1: Exemplo de execução (para pontos sucessivos) com a melhor geometria

Gráfico 1: Média dos tempos de execução (para pontos sucessivos)

GPU											
blockDimX	1	2	4	8	16	32	64	128	256	512	1024
T1	1,576	0,816	0,436	0,247	0,131	0,072	0,066	0,066	0,065	0,071	0,074
T2	1,575	0,817	0,435	0,247	0,131	0,072	0,066	0,066	0,065	0,072	0,074
T3	1,576	0,817	0,436	0,247	0,131	0,072	0,066	0,066	0,066	0,071	0,074
T4	1,577	0,816	0,435	0,247	0,131	0,072	0,066	0,066	0,065	0,071	0,074
T5	1,575	0,816	0,436	0,247	0,131	0,072	0,066	0,066	0,065	0,071	0,074
Média	1,576	0,816	0,436	0,247	0,131	0,072	0,066	0,066	0,065	0,071	0,074
DESVP	0,0008	0,0005	0,0005	0,0000	0,0000	0,0000	0,0000	0,0000	0,0004	0,0004	0,0000

Tabela 1: Tempos de execução do GPU (para pontos sucessivos)

CPU											
blockDimX	1	2	4	8	16	32	64	128	256	512	1024
T1	4,258	4,239	4,303	4,266	4,258	4,26	4,268	4,25	4,262	4,254	4,273
T2	4,271	4,248	4,219	4,234	4,275	4,264	4,249	4,248	4,279	4,242	4,254
T3	4,27	4,274	4,256	4,264	4,215	4,271	4,254	4,25	4,251	4,249	4,242
T4	4,233	4,224	4,234	4,288	4,243	4,256	4,267	4,297	4,283	4,269	4,244
T5	4,23	4,232	4,233	4,277	4,28	4,235	4,253	4,261	4,281	4,262	4,276
Média	4,252	4,243	4,249	4,266	4,254	4,257	4,258	4,261	4,271	4,255	4,258
DESVP	0,0198	0,0193	0,0330	0,0202	0,0263	0,0136	0,0087	0,0207	0,0140	0,0106	0,0159

Tabela 2: Tempos de execução do CPU (para pontos sucessivos)

## Exercício 2 - Resultados

MÉDIA DOS TEMPOS DE EXECUÇÃO

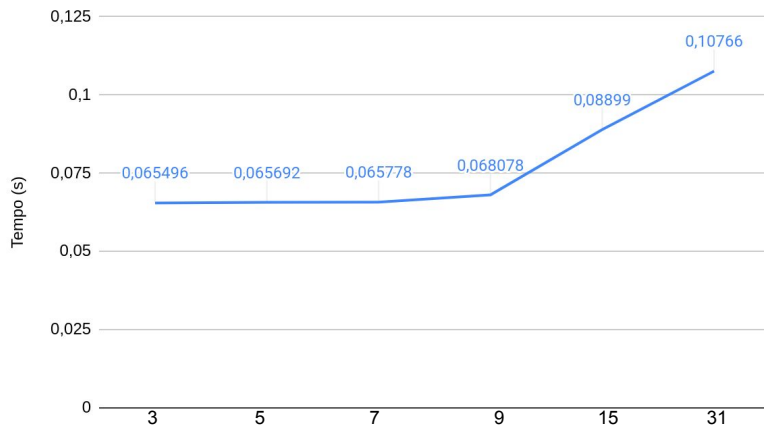


Gráfico 2: Média dos tempos de execução do GPU (para pontos separados por uma distância fixa)

MÉDIA DOS TEMPOS DE EXECUÇÃO

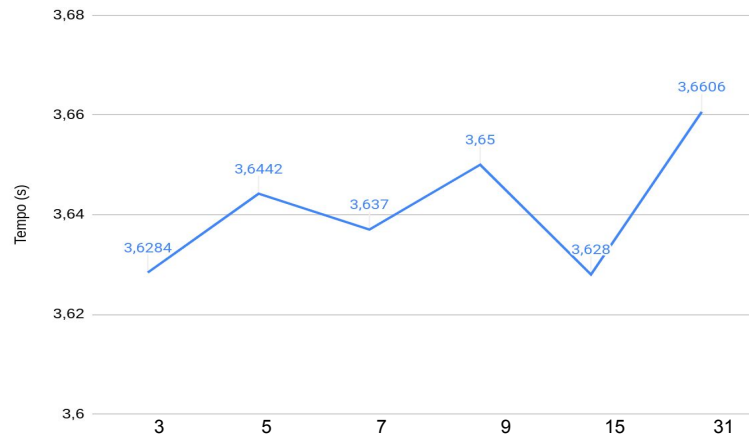


Gráfico 3: Média dos tempos de execução do CPU (para pontos separados por uma distância fixa)

```
./cccCuda Starting...
Using Device 0: GeForce GTX 1660 Ti
Total samples data size: 524288
The initialization of host data took 1.724e-03 seconds
The transfer of 1572864 bytes from the host to the device took 2.503e-04 seconds
The CUDA kernel <<<(256,1,1), (256,1,1)>>> took 6.551e-02 seconds to run
The transfer of 524288 bytes from the device to the host took 2.401e-04 seconds
The cpu kernel took 3.634e+00 seconds to run (single core)
All is well!
```

Figura 2: Exemplo de execução (para pontos separados por uma distância fixa) com a melhor geometria e a melhor distância

GPU						
DISTANCE	3	5	7	9	15	31
T1	0,06552	0,06566	0,06577	0,06813	0,08892	0,1078
T2	0,06547	0,06571	0,0658	0,06804	0,08893	0,1073
T3	0,0655	0,06566	0,06576	0,06809	0,08897	0,108
T4	0,0655	0,06571	0,06579	0,06804	0,08906	0,1076
T5	0,06549	0,06572	0,06577	0,06809	0,08907	0,1076
Média	0,065496	0,065692	0,065778	0,068078	0,08899	0,10766
DESVP	0,000018	0,000029	0,000016	0,000038	0,000071	0,000261

Tabela 3: Tempos de execução do GPU (para pontos separados por uma distância fixa)

CPU						
DISTANCE	3	5	7	9	15	31
T1	3,643	3,678	3,627	3,665	3,625	3,649
T2	3,628	3,649	3,635	3,677	3,644	3,665
T3	3,622	3,628	3,63	3,65	3,621	3,653
T4	3,631	3,63	3,669	3,636	3,625	3,658
T5	3,618	3,636	3,624	3,622	3,625	3,678
Média	3,6284	3,6442	3,637	3,65	3,628	3,6606
DESVP	0,009607	0,020596	0,018344	0,021989	0,009110	0,011415

Tabela 4: Tempos de execução do CPU (para pontos separados por uma distância fixa)



## Conclusão

- Através dos dados obtidos das várias execuções é possível concluir que a melhor geometria é quando o `blockDimx` é  $2^8$ .
- No segundo exercício, como é possível observar quanto maior a distância maior o tempo de execução, isto deve-se ao acesso aos dados dos arrays que estão em memória.
- "Is it worthwhile to use the GPU to run this kind of problems?"
  - Sim, vale. Analisando os tempos de execução, é notório que os do GPU são sempre inferiores aos do CPU, mesmo para a pior geometria. Isto deve-se ao facto dos cálculos serem feitos em paralelo/simultâneo pelas várias *threads*.