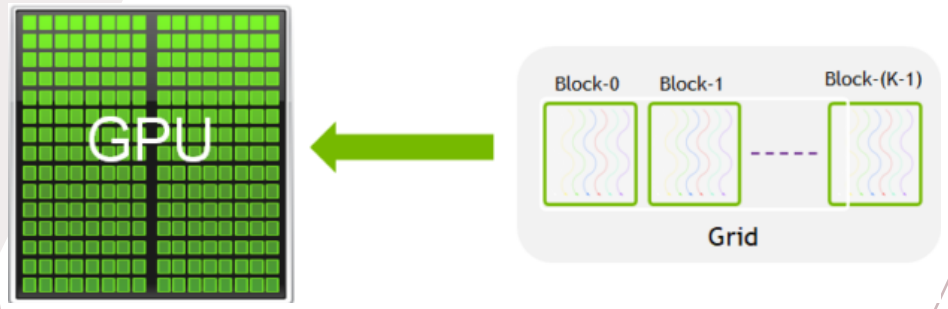
ARQUITETURAS DE ALTO DESEMPENHO

TRABALHO2 : ORDENAR SEQUÊNCIAS DE VALORES

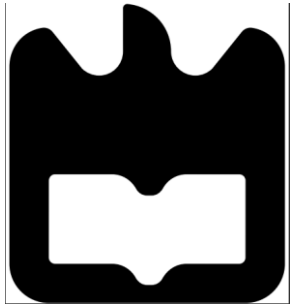


UNIVERSIDADE AVEIRO

DETI: DEPARTAMENTO DE ELETRÓNICA E TELECOMUNICAÇÕES

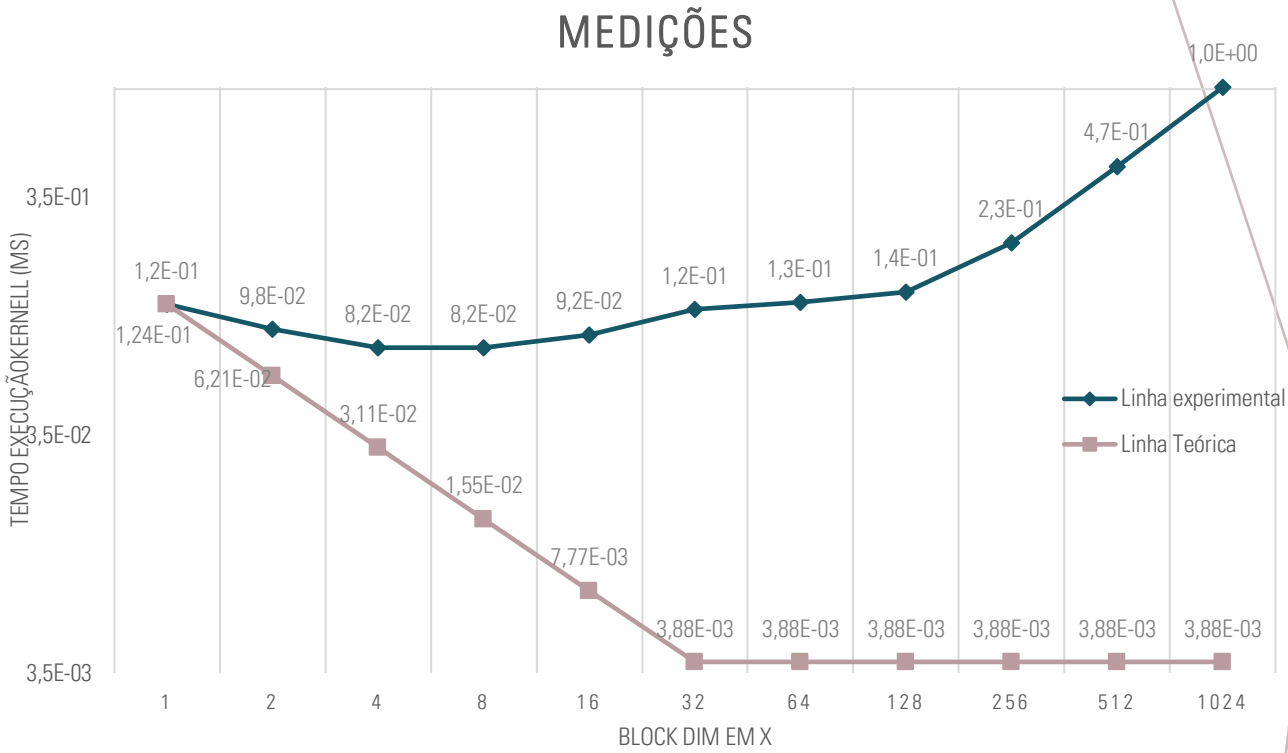
MARTA OLIVEIRA
97613(MARTA.ALEX@UA.PT)

BRUNO SILVA
97931(BRUNOSILVA16@UA.PT)



INCORDERROW ANÁLISE

- A partir das 16 threads por bloco até às 128 threads o tempo de execução aumenta de forma ligeira, sendo que a partir das 256 threads aumenta de forma exponencial.
- A organização dos dados leva a um acesso pouco eficiente por múltiplas threads. As threads acedem a áreas não contínuas da memória o que leva a uma maior ocorrência de cache misses.
- O benefício de correr mais threads em paralelo só é notado até às 8 threads por bloco, a partir daí este já não compensa em relação aos cache misses.
- Após análise, obtivemos 2 configurações que forneceram a melhor performance: **4 e 8 threads por bloco**.



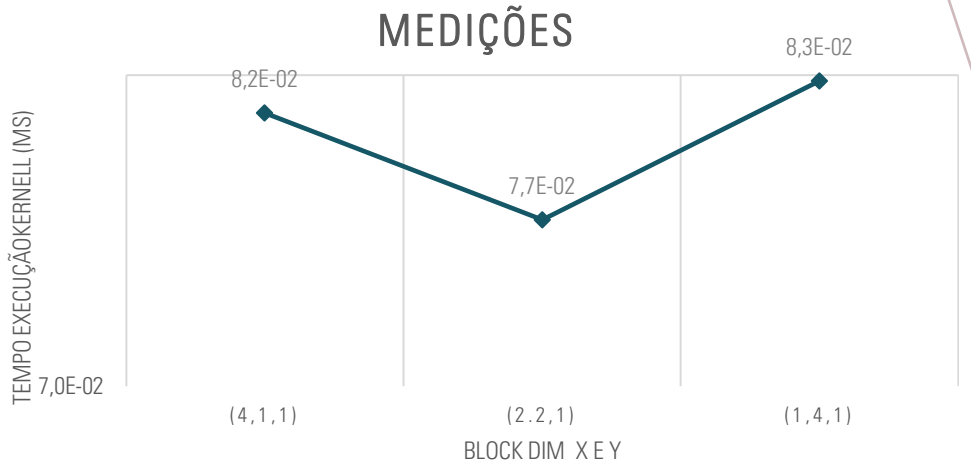
Variação do valor das threads por bloco

INCORDERROW ANÁLISE

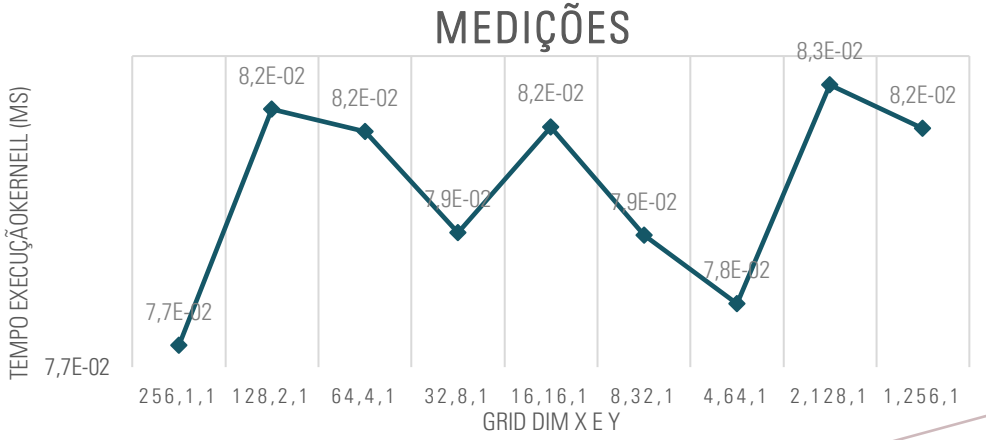
- O melhor resultado no "desempate" entre as 4 e 8 threads por bloco foi obtido para as 4 threads, organizadas em (2, 2).
- O melhor resultado obtido após a variação da grid dos blocos foi a configuração inicial (256 blocos em x) .
- **Melhor configuração obtida**: (256, 1, 1), (2, 2, 1).
- Média de tempo de execução do CPU: 6,2E-01 s

Conclusões

- Apesar da memória não estar mapeada da melhor forma e de o tempo de execução da GPU não ter diminuído da maneira prevista teoricamente, a performance da GPU foi superior à do CPU.



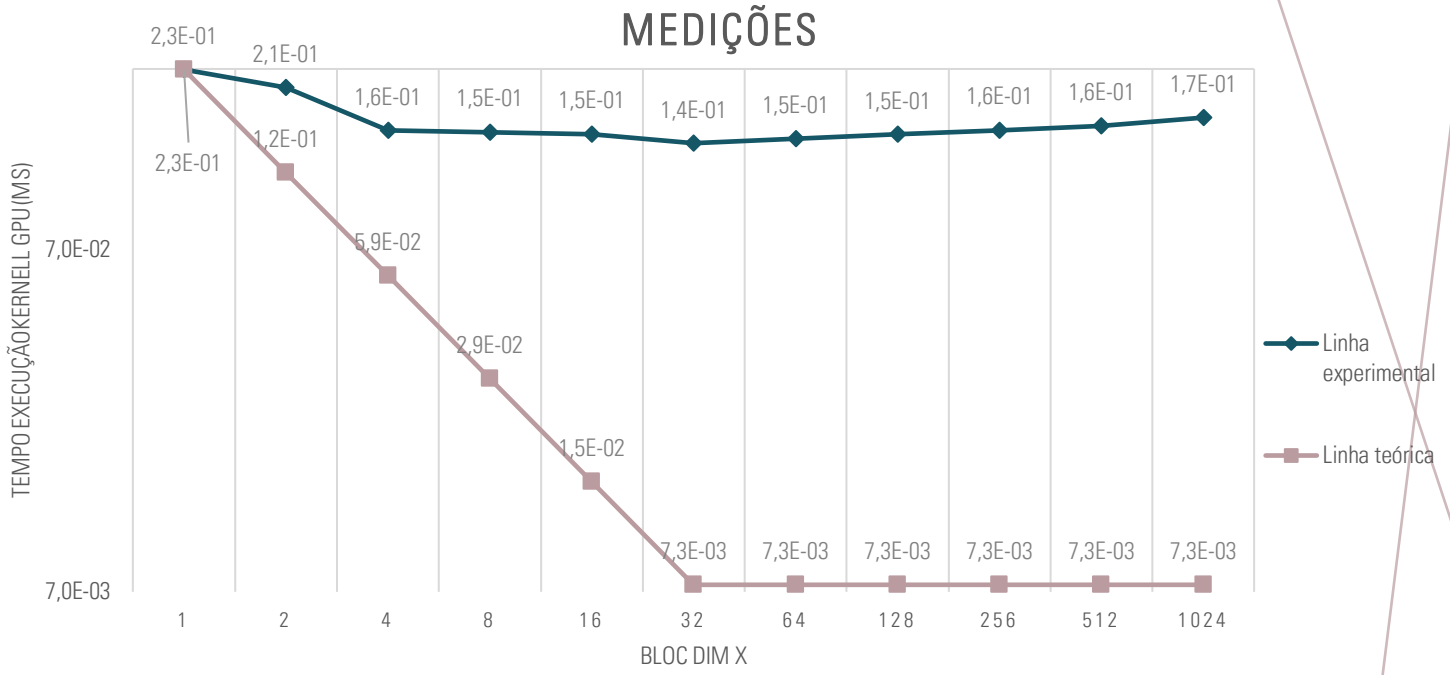
Variação da grid das threads por bloco



Variação da grid dos blocos

INCORDERCOLUMN ANÁLISE

- Em geral, embora não exista uma variação muito acentuada nos tempos de execução obtidos, conseguimos ainda assim observar uma diminuição mais acentuada do tempo entre as 2 e as 4 threads por bloco
- Melhor resultado obtido para **32 threads por bloco.** Tendo em conta que este valor corresponde ao tamanho das warps da GPU utilizada, concluímos que quando existe uma ocupação total das warps é alcançada uma melhor performance.
- Como neste caso as threads vão aceder a áreas contínuas de memória, sempre que uma thread aceder a um novo valor da sequência, existe uma maior probabilidade de esse elemento não estar presente na cache pois cada elemento da sequência está mais distante.
- O tempo de execução manteve-se constante dado que é introduzido um atraso por cada acesso à sequência que necessita que o valor seja retirado da memória principal, sendo este aproveitado pelo core para executar outra warp. Esta é também a razão para o tempo de execução não diminuir como foi teoricamente previsto.



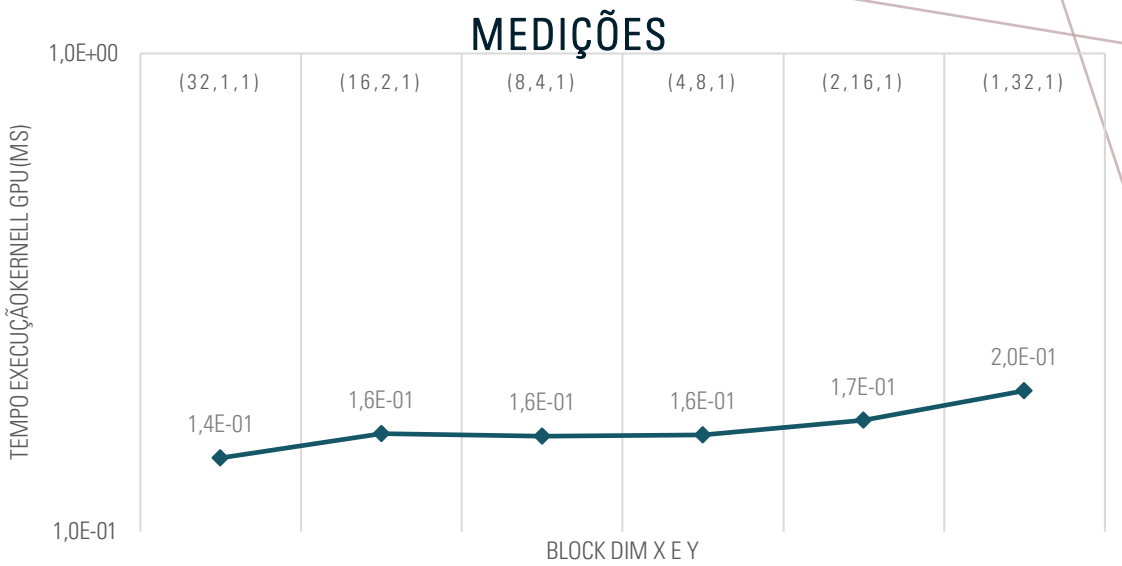
Variação do valor das threads por bloco

INCORDERCOLUMN ANÁLISE

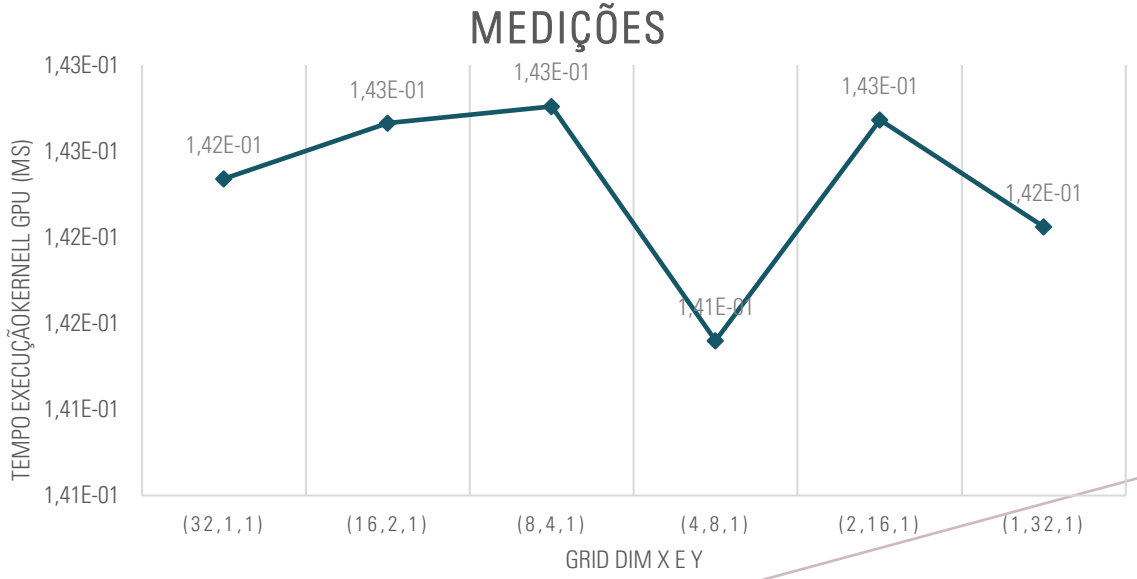
- Após concluirmos que 32 threads por bloco nos forneceu a melhor performance, variámos a configuração da grid das threads por bloco, sendo que a melhor performance continuou a ser a das 32 threads por bloco.
- O melhor resultado obtido após a variação da grid dos blocos foi a configuração (4, 8)
- **Melhor configuração obtida**: (4,8,1),(32,1,1)
- Média de tempo de execução do CPU: 9,05 s

Conclusões

- A performance da GPU foi superior à do CPU, sendo isto justificado pelo facto do CPU ordenar as sequências de valores de forma sequencial, e como estes valores não são contínuos na memória existe um número elevado de cache misses.



Variação da grid das threads por bloco



Variação da grid dos blocos