



# ARQUITETURAS DE ALTO DESEMPENHO

## ASSIGNMENT 2 - PROOF TO WORK

**Turma 1**

Rui Miguel Oliveira - 89216

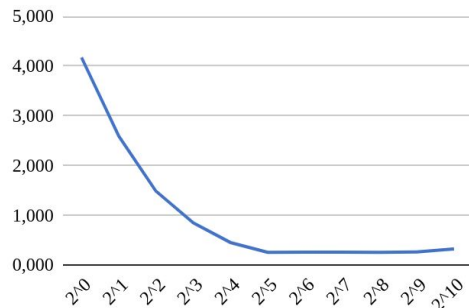
# POW24.cu - Optimização

GRID		BLOCK		TEMPO MÉDIO	DESVIO PADRÃO
X	Y	X	Y		
2^29	2^0	2^0	2^0	4,174	0,007
2^28	2^0	2^1	2^0	2,596	0,006
2^27	2^0	2^2	2^0	1,489	0,002
2^26	2^0	2^3	2^0	0,853	0,002
2^25	2^0	2^4	2^0	0,456	0,006
2^24	2^0	2^5	2^0	0,258	0,003
2^23	2^0	2^6	2^0	0,262	0,004
2^22	2^0	2^7	2^0	0,263	0,003
2^21	2^0	2^8	2^0	0,261	0,002
2^20	2^0	2^9	2^0	0,269	0,004
2^19	2^0	2^10	2^0	0,330	0,004

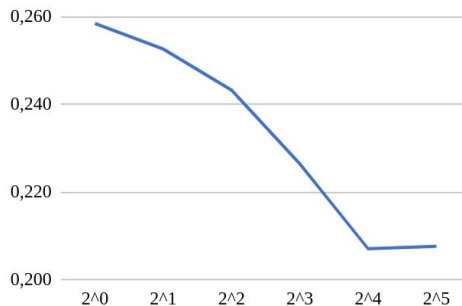
GRID		BLOCK		TEMPO MÉDIO	DESVIO PADRÃO
X	Y	X	Y		
2^24	2^0	2^5	2^0	0,258	0,003
2^23	2^0	2^5	2^1	0,253	0,005
2^22	2^0	2^5	2^2	0,243	0,004
2^21	2^0	2^5	2^3	0,226	0,000
2^20	2^0	2^5	2^4	0,207	0,000
2^19	2^0	2^5	2^5	0,208	0,000

GRID		BLOCK		TEMPO MÉDIO	DESVIO PADRÃO
X	Y	X	Y		
2^20	2^0	2^5	2^4	0,20706	0,000
2^19	2^1	2^5	2^4	0,16665	0,000
2^18	2^2	2^5	2^4	0,16665	0,000
2^17	2^3	2^5	2^4	0,16662	0,000
2^16	2^4	2^5	2^4	0,16665	0,000
2^15	2^5	2^5	2^4	0,16665	0,000
2^14	2^6	2^5	2^4	0,16669	0,000
2^13	2^7	2^5	2^4	0,16669	0,000
2^12	2^8	2^5	2^4	0,16673	0,000
2^11	2^9	2^5	2^4	0,16683	0,000
2^10	2^10	2^5	2^4	0,16671	0,000

VARIAÇÃO DO TEMPO EM FUNÇÃO DE BLOCK X



VARIAÇÃO DO TEMPO EM FUNÇÃO DE BLOCK Y



VARIAÇÃO DO TEMPO EM FUNÇÃO DA GRID



# POW24.cu - Conclusões

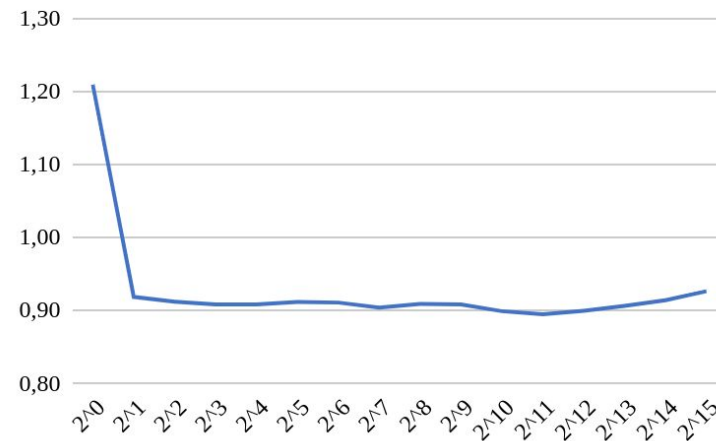
---

- Melhor configuração  $\langle (17, 3), (5, 4) \rangle$
- Tempo de execução na GPU é quase 900 vezes mais rápido do que no CPU:  
Tempo execução médio CPU - 152.57s.  
Tempo execução médio CUDA - 0,17s (depois da optimização).
- A variação entre os valores de X e Y da *grid* têm pouco impacto.
- Ocorrem muitas falhas de cache.  
Uma grande parte dos bytes carregados nas caches internos de cada thread não são usados e precisam ser transferidos novamente.

# POW27.cu - Optimização

GRID		BLOCK		TEMPO MÉDIO	DESVIO PADRÃO
X	Y	X	Y		
2 <sup>23</sup>	2 <sup>0</sup>	2 <sup>5</sup>	2 <sup>4</sup>	1,21	0,01
2 <sup>22</sup>	2 <sup>1</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,92	0,01
2 <sup>21</sup>	2 <sup>2</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>20</sup>	2 <sup>3</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>19</sup>	2 <sup>4</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,00
2 <sup>18</sup>	2 <sup>5</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>17</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>16</sup>	2 <sup>7</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,90	0,01
2 <sup>15</sup>	2 <sup>8</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>14</sup>	2 <sup>9</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>13</sup>	2 <sup>10</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,90	0,01
2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,89	0,01
2 <sup>11</sup>	2 <sup>12</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,90	0,00
2 <sup>10</sup>	2 <sup>13</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>09</sup>	2 <sup>14</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,91	0,01
2 <sup>08</sup>	2 <sup>15</sup>	2 <sup>5</sup>	2 <sup>4</sup>	0,93	0,00

VARIAÇÃO DO TEMPO EM FUNÇÃO DA GRID



# POW27.cu - Conclusões

---

- Melhor configuração  $\langle (12, 11), (5, 4) \rangle$
- Tempo de execução na GPU é cerca de 176 vezes mais rápido do que no CPU:  
Tempo execução médio CPU - 156.69.  
Tempo execução médio CUDA - 0,89s (depois da optimização).
- Para evitar computação desnecessária, assim que um valor de  $n[0]$  é aceitável, o programa termina.
- O PoW24 é mais rápido com o compromisso de não obter um token para alguns valores, pois a probabilidade de obter um valor é  $1 - (1 - 2^{-24})^k = 0.63212$ , o PoW27 é mais preciso com uma probabilidade de 0.99966 de obter um token.