

ESERCITAZIONE 2

Caratteristiche macchina:

CPU: Intel i7-12700H

RAM: 16 GB, 4800 MHz

**** i valori nelle tabelle sono la media del risultato di tre esecuzioni.**

ALGORITMO SERIALE:

Processori = 1

N	tempo (s)
1.000x1.000	0,00346
4.000x4.000	0,05993
8.000x8.000	0,23516
10.000x10.000	0,35294
12.000x12.000	0,52296

1° STRATEGIA PRODOTTO MATRICE-VETTORE (suddivisione a righe)

Processori = 2

N	tempo (s)	Sp	Ep (%)
1.000x1.000	0,00177	1,95	97,74
4.000x4.000	0,03063	1,96	97,83
8.000x8.000	0,11316	2,08	103,91
10.000x10.000	0,18203	1,94	96,95
12.000x12.000	0,28733	1,82	91,00

Processori = 4

N	tempo (s)	Sp	Ep (%)
1.000x1.000	0,00088	3,93	98,30
4.000x4.000	0,01456	4,12	102,90
8.000x8.000	0,05644	4,17	104,16
10.000x10.000	0,09045	3,90	97,55
12.000x12.000	0,1348	3,88	96,99

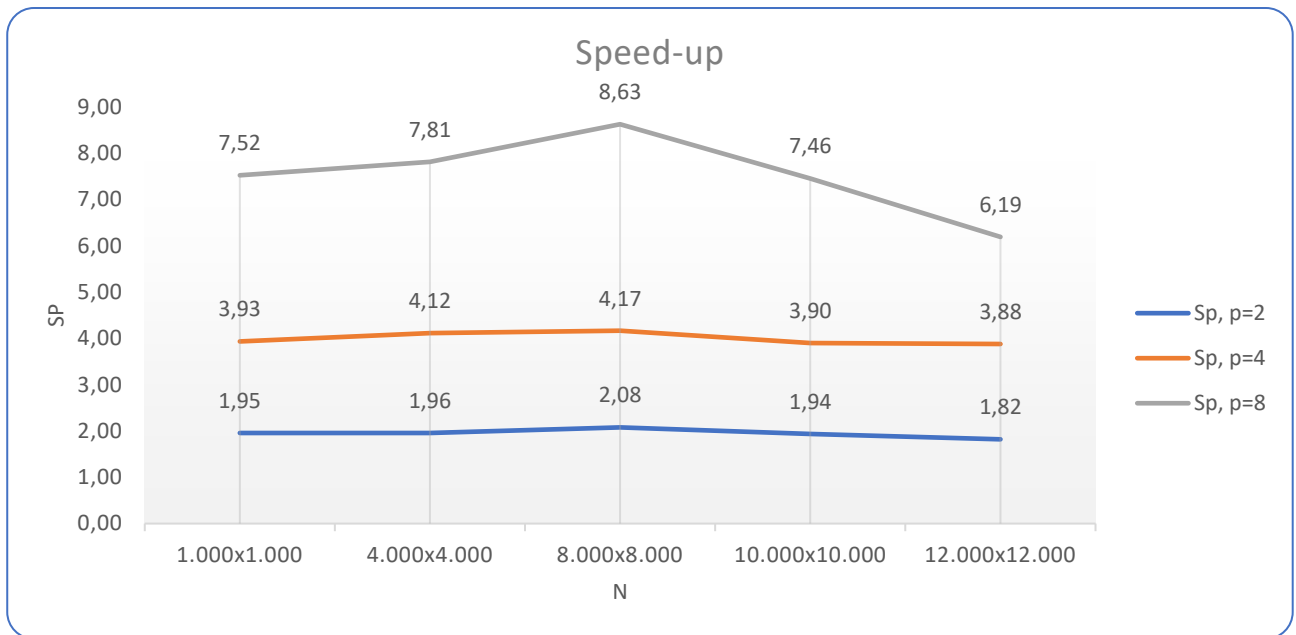
Processori = 8

N	tempo (s)	Sp	Ep (%)
1.000x1.000	0,00046	7,52	94,02
4.000x4.000	0,00767	7,81	97,67
8.000x8.000	0,02726	8,63	107,83
10.000x10.000	0,04734	7,46	93,19
12.000x12.000	0,08445	6,19	77,41

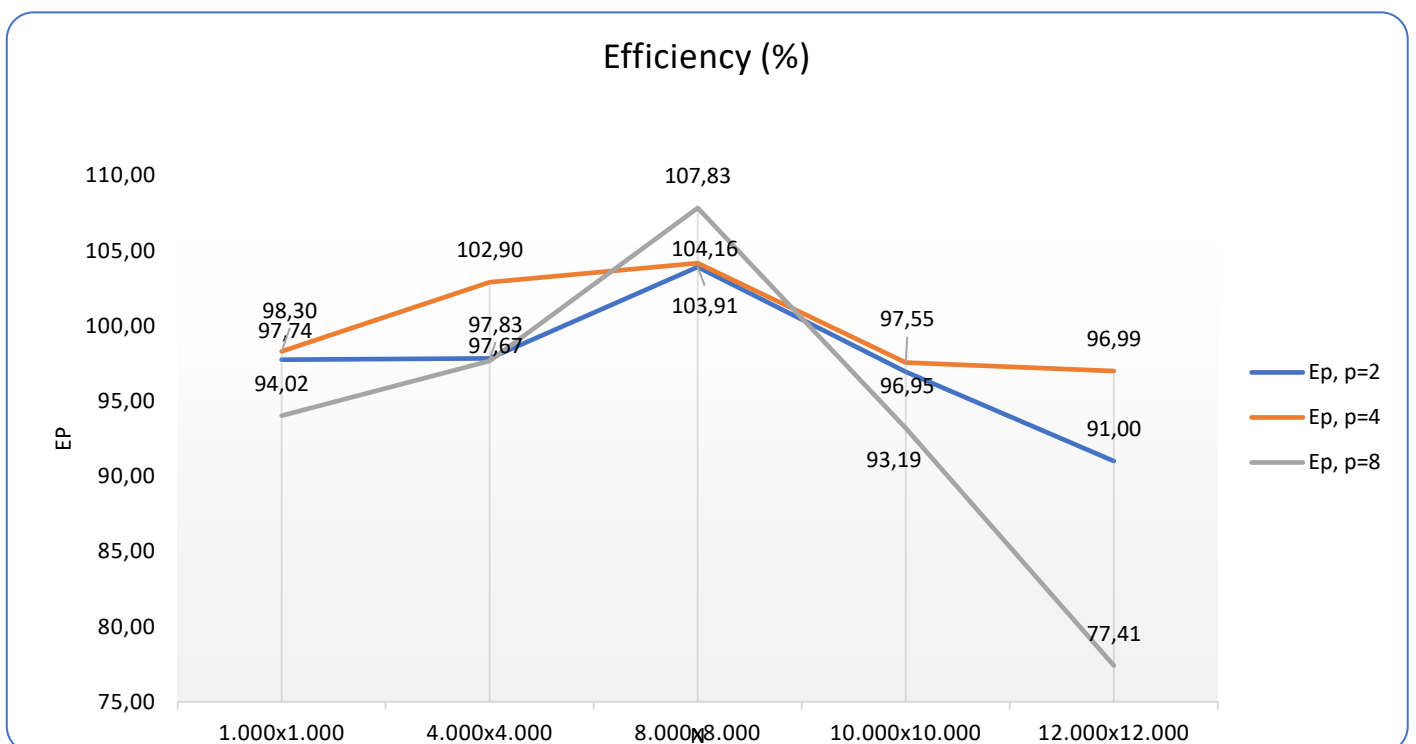
Secondo la *legge di Amhdal*:

se consideriamo p fissato, al crescere della dimensione n del problema la parte sequenziale tende a 0 e la parte parallela aumenta, **Speed-up** ed **Efficienza** tendono ai loro valori **ideali**.

Dai dati risulta che lo speed-up aumenta fino ad una dimensione pari a 8000×8000 ottenendo uno speed-up super lineare, ma successivamente diminuisce man mano che la dimensione del problema aumenta.



Se analizziamo l'efficienza in ogni suo punto, all'interno del grafico, l'ideale sarebbe utilizzare 4 processori in quanto limita superiormente, nella quasi totalità delle volte, gli altri valori. Si evidenzia, inoltre, che solo nel caso in cui la dimensione del problema sia pari a 8000x8000 risulta essere più efficiente l'utilizzo di 8 processori.



2° STRATEGIA PRODOTTO MATRICE-VETTORE (suddivisione a colonne)

Processori = 2

N	tempo (s)	Sp	Ep
1.000x1.000	0,00242	1,43	71,49
4.000x4.000	0,03956	1,51	75,75
8.000x8.000	0,18842	1,25	62,40
10.000x10.000	0,42229	0,84	41,79

Processori = 4

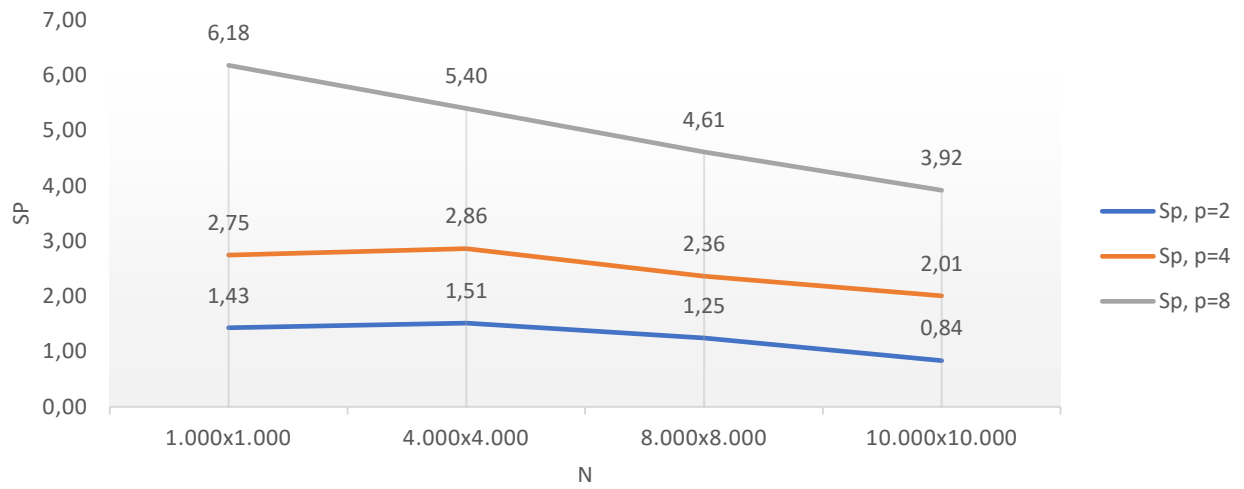
N	tempo (s)	Sp	Ep
1.000x1.000	0,00126	2,75	68,65
4.000x4.000	0,02094	2,86	71,55
8.000x8.000	0,0995	2,36	59,09
10.000x10.000	0,17575	2,01	50,20

Processori = 8

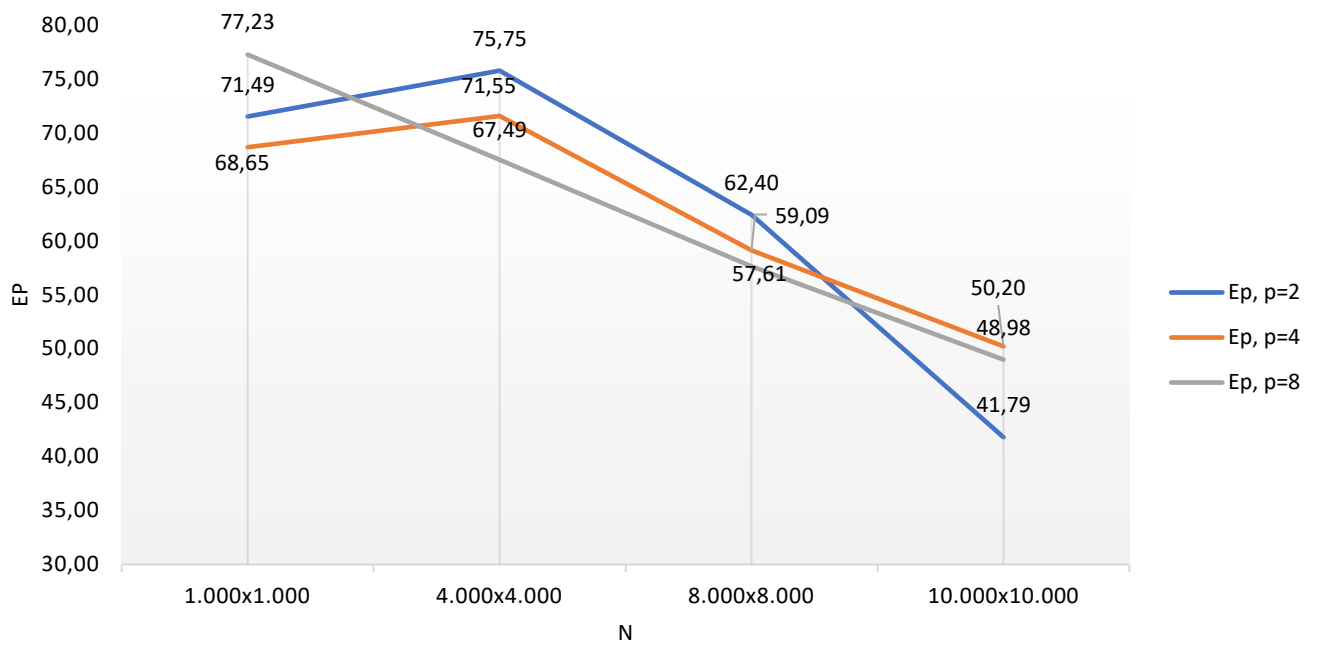
N	tempo (s)	Sp	Ep (%)
1.000x1.000	0,00056	6,18	77,23
4.000x4.000	0,0111	5,40	67,49
8.000x8.000	0,05102	4,61	57,61
10.000x10.000	0,09008	3,92	48,98

Nella seconda strategia riscontriamo un peggioramento non solo dello speed-up ma anche dall'efficienza dell'algoritmo.

Speed-up



Efficiency (%)



3° STRATEGIA PRODOTTO MATRICE-VETTORE (suddivisione a righe e colonne)

Processori = 2

N	tempo (s)	Sp	Ep
1.000x1.000	0,00182	1,90	95,05
4.000x4.000	0,02968	2,02	100,96
8.000x8.000	0,11496	2,05	102,28
10.000x10.000	0,17885	1,97	98,67

Processori = 4

N	tempo (s)	Sp	Ep
1.000x1.000	0,00087	3,98	99,43
4.000x4.000	0,01695	3,54	88,39
8.000x8.000	0,05908	3,98	99,51
10.000x10.000	0,09212	3,83	95,78

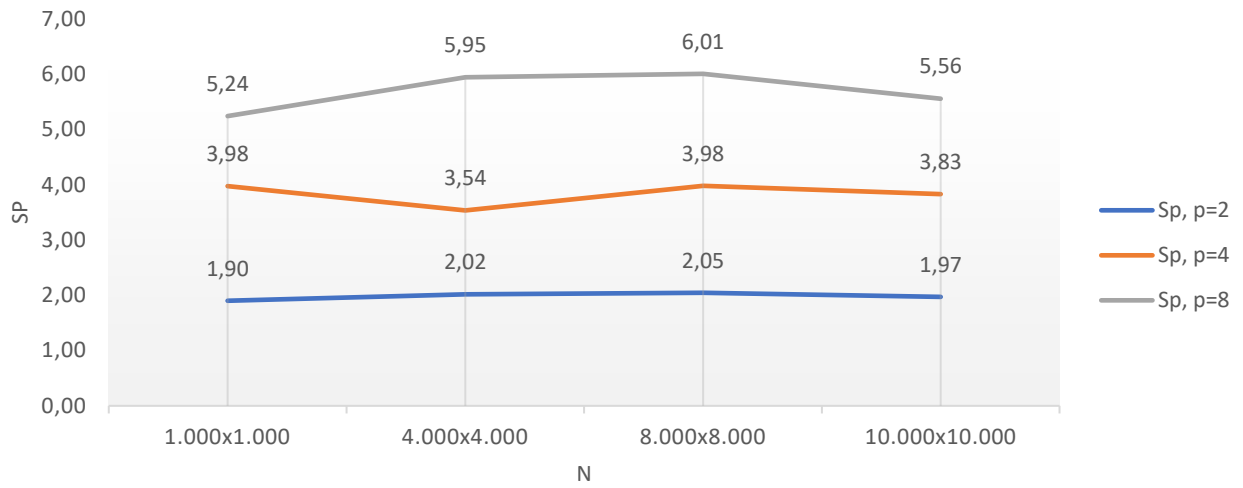
Processori = 8

N	tempo (s)	Sp	Ep (%)
1.000x1.000	0,00066	5,24	65,53
4.000x4.000	0,01008	5,95	74,32
8.000x8.000	0,03915	6,01	75,08
10.000x10.000	0,0635	5,56	69,48

Nella **3 strategia** notiamo sicuramente un netto miglioramento rispetto alla seconda. Invece confrontandolo con la **1 strategia** notiamo che da 2 fino a 4 processori l'efficienza risulta quasi la medesima, ma per 8 processori la 3 strategia risulta peggiore.

Nella terza strategia abbiamo un miglioramento rispetto alla seconda. Infatti, per la legge di Amdahl fissando il numero di processori e aumentando la taglia del problema otteniamo degli speed-up sempre migliori. Anche qui notiamo un piccolo picco per i valori 8000x8000 raggiungendo lo speed-up ideale.

Speed-up



Efficiency (%)

