

# ESERCITAZIONE 1

## **Caratteristiche macchina:**

**CPU:** AMD Ryzen 5 5500

**RAM:** 16 GB, 3600 MHz

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

## **ALGORITMO SERIALE:**

**Processori = 1**

<b>N</b>	<b>tempo (s)</b>
10.000.000	0,028744
20.000.000	0,057538
40.000.000	0,115419
8,00E+07	0,231205
1,00E+08	0,289393

## 2° STRATEGIA SOMMA DI N NUMERI

Processori = 2

N	tempo (s)	Sp	Ep (%)
10.000.000	0,014842	1,94	96,83
20.000.000	0,02907	1,98	98,96
40.000.000	0,058741	1,96	98,24
8,00E+07	0,128175	1,80	90,19
1,00E+08	0,146005	1,98	99,10

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=2$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 2 e l'efficienza tende ad 1 (=100%).

Processori = 4

N	tempo (s)	Sp	Ep (%)
10.000.000	0,007692	3,74	93,42
20.000.000	0,014457	3,98	99,50
40.000.000	0,030481	3,79	94,66
8,00E+07	0,058872	3,93	98,18
1,00E+08	0,074703	3,87	96,85

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=4$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 4 e l'efficienza tende ad 1 (=100%).

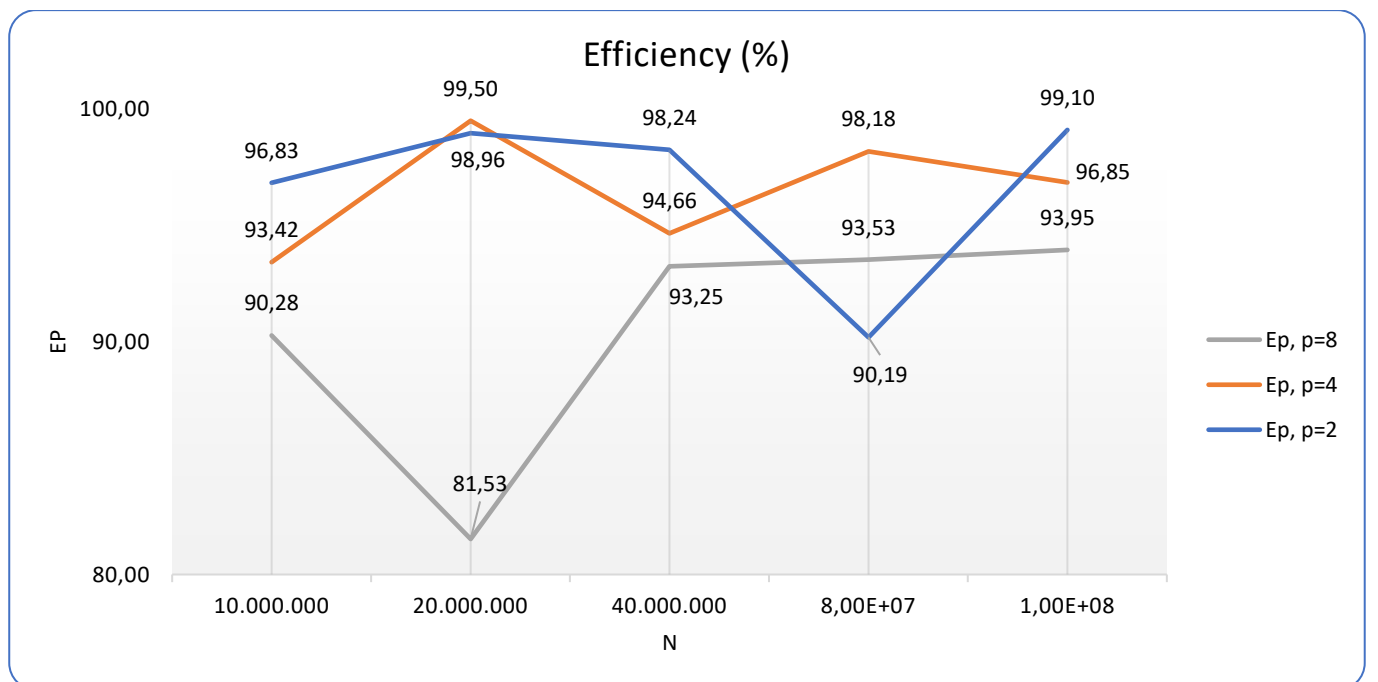
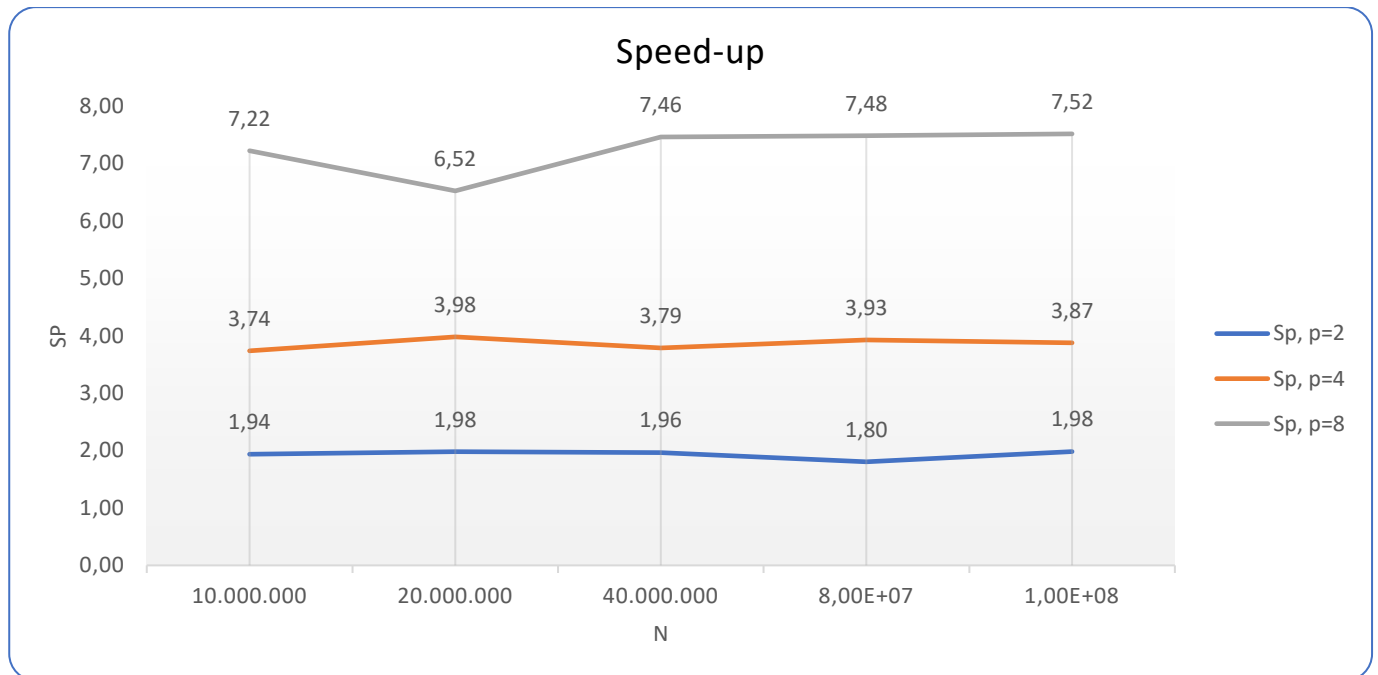
Processori = 8

N	tempo (s)	Sp	Ep (%)
10.000.000	0,00398	7,22	90,28
20.000.000	0,008822	6,52	81,53
40.000.000	0,015472	7,46	93,25
8,00E+07	0,0309	7,48	93,53
1,00E+08	0,038504	7,52	93,95

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=8$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 8 e l'efficienza tende ad 1. (100%).

Come possiamo notare lo speed-up migliora ma in realtà la parte parallela diminuisce. Infatti, se andiamo a vedere l'Efficienza, che appunto misura quanto l'algoritmo sfrutta il parallelismo del calcolatore, notiamo che diminuisce sempre di più all'aumentare dei processori.



### 3° STRATEGIA SOMMA DI N NUMERI

Processori = 2

N	tempo (s)	Sp	Ep(%)
10.000.000	0,014589	1,97	98,51
20.000.000	0,02968	1,94	96,93
40.000.000	0,058015	1,99	99,47
8,00E+07	0,115933	1,99	99,71
1,00E+08	0,145933	1,98	99,15

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=2$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 2 e l'efficienza tende ad 1. (100%).

Processori = 4

N	tempo (s)	Sp	Ep(%)
10.000.000	0,007588	3,79	94,70
20.000.000	0,015396	3,74	93,43
40.000.000	0,029634	3,89	97,37
8,00E+07	0,061683	3,75	93,71
1,00E+08	0,075899	3,81	95,32

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=4$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 4 e l'efficienza tende ad 1. (100%).

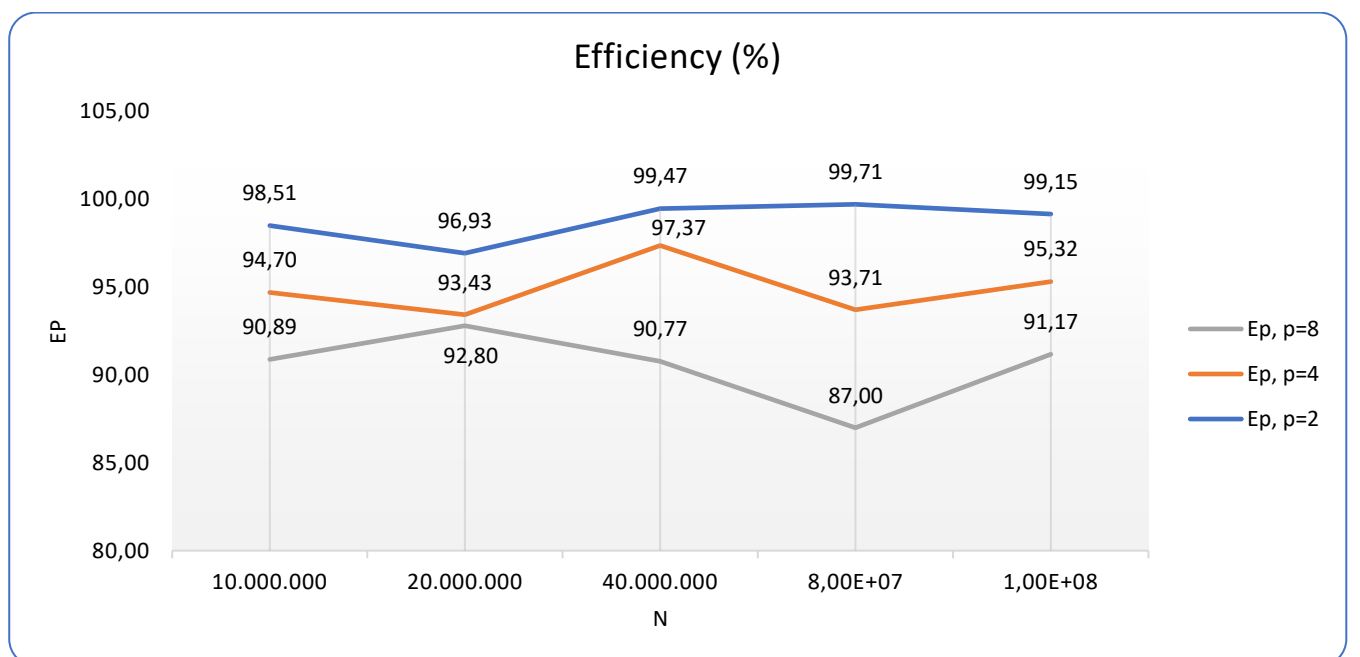
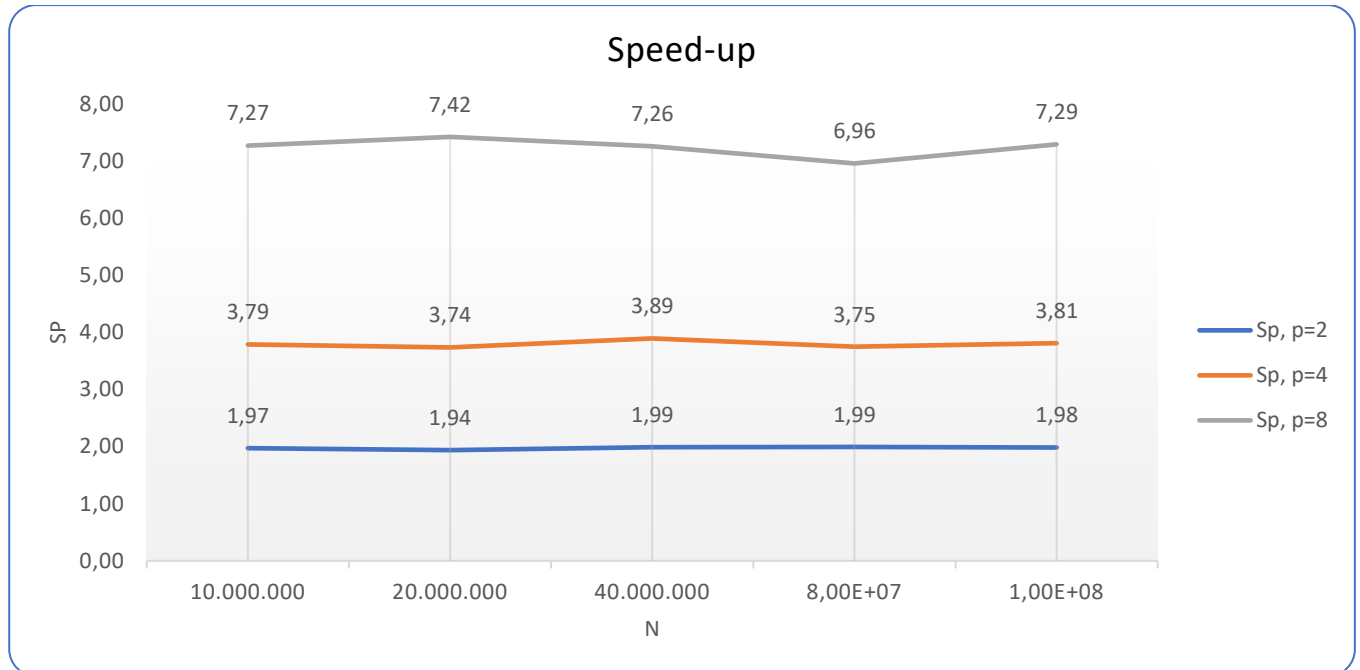
Processori = 8

N	tempo (s)	Sp	Ep (%)
10.000.000	0,003953	7,27	90,89
20.000.000	0,00775	7,42	92,80
40.000.000	0,015894	7,26	90,77
8,00E+07	0,033221	6,96	87,00
1,00E+08	0,039677	7,29	91,17

Secondo la *legge di Amhdal*:

se consideriamo  $p$  fissato ( $p=8$ ), 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**. Come possiamo notare anche qui lo speed-up tende al suo valore ideale 8 e l'efficienza tende ad 1. (100%).

Come possiamo notare lo speed-up migliora ma in realtà la parte parallela diminuisce. Infatti, se andiamo a vedere l'Efficienza, che appunto misura quanto l'algoritmo sfrutta il parallelismo del calcolatore, notiamo che diminuisce sempre di più all'aumentare dei processori.



## CONSIDERAZIONI FINALI:

Quali delle due strategie è meglio utilizzare?

Strategia 2

Sp	Ep
1,94	96,83
1,98	98,96
1,96	98,24
1,80	90,19
1,98	99,10
Sp	Ep(%)
3,74	93,42
3,98	99,50
3,79	94,66
3,93	98,18
3,87	96,85
Sp	Ep (%)
7,22	90,28
6,52	81,53
7,46	93,25
7,48	93,53
7,52	93,95

Strategia 3

Sp	Ep
1,97	98,51
1,94	96,93
1,99	99,47
1,99	99,71
1,98	99,15
Sp	Ep
3,79	94,70
3,74	93,43
3,89	97,37
3,75	93,71
3,81	95,32
Sp	Ep (%)
7,27	90,89
7,42	92,80
7,26	90,77
6,96	87,00
7,29	91,17



Come si può notare la seconda strategia risulta migliore dal punto di vista dello Speed-up e dell'Efficienza. Questo perché effettua meno comunicazioni e meno operazioni. Alla fine, il numero di passi resta invariato e la differenza è veramente piccola, quindi la scelta di quale delle due strategie fa riferimento al tipo di problema che vogliamo risolvere.

In questo caso preferiamo la seconda strategia, in quanto il nostro obiettivo è solo quello di restituire e stampare il risultato finale della somma. Se avessimo dovuto utilizzare questa somma per altri calcoli e quindi avevamo bisogno che ogni processore avesse la somma calcolata avremmo preferito la terza strategia.