Lab. 5 - Clustering k-means parallelo

Componenti gruppo:

- **Federico Caldart , matricola: 1211144**
- **Stefano Panozzo, matricola: 1211143**
- **Davide Zago, matricola: 1211260**

Domanda 1

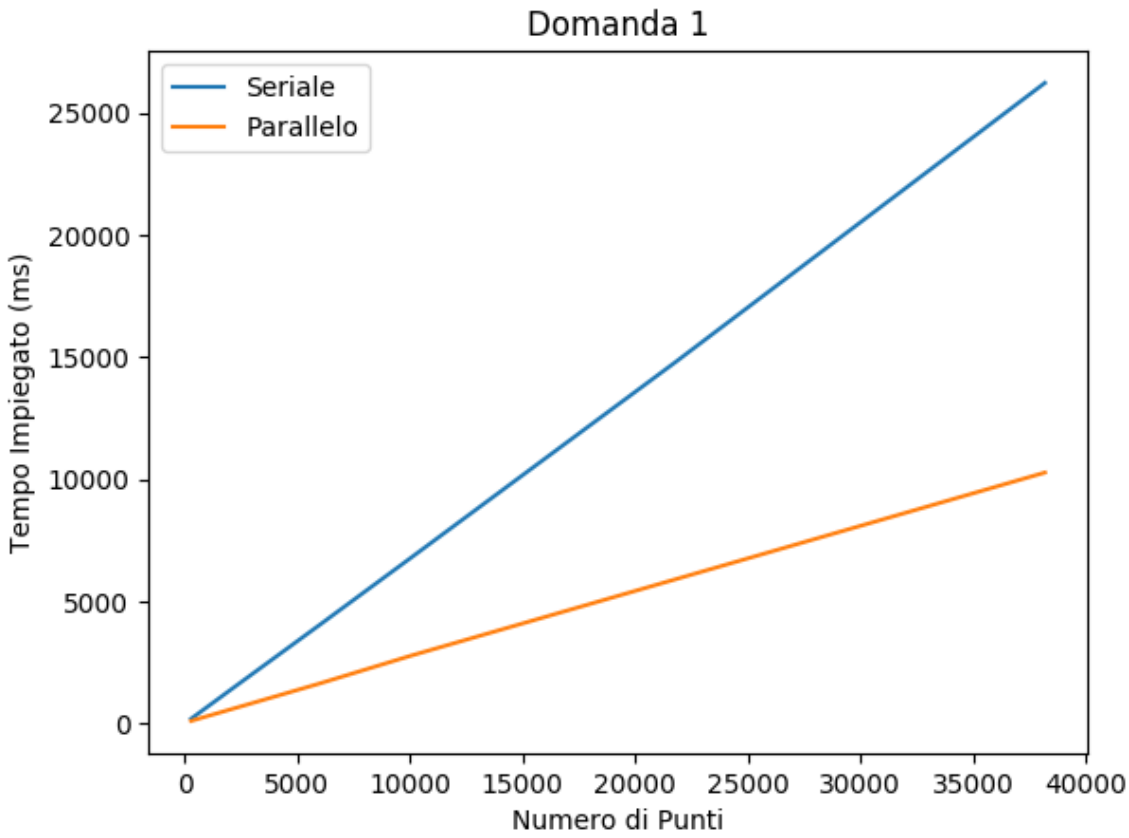

Fig.1: Confronto dei tempi di calcolo seriale e parallelo al variare del numero di punti, con numero di cluster costante a 50 e numero di iterazioni costante a 100; si nota come la versione parallela sia di gran lunga più veloce della versione seriale: entrambe le rette crescono in maniera lineare, ma quella relativa al parallelo ha una pendenza visivamente minore.

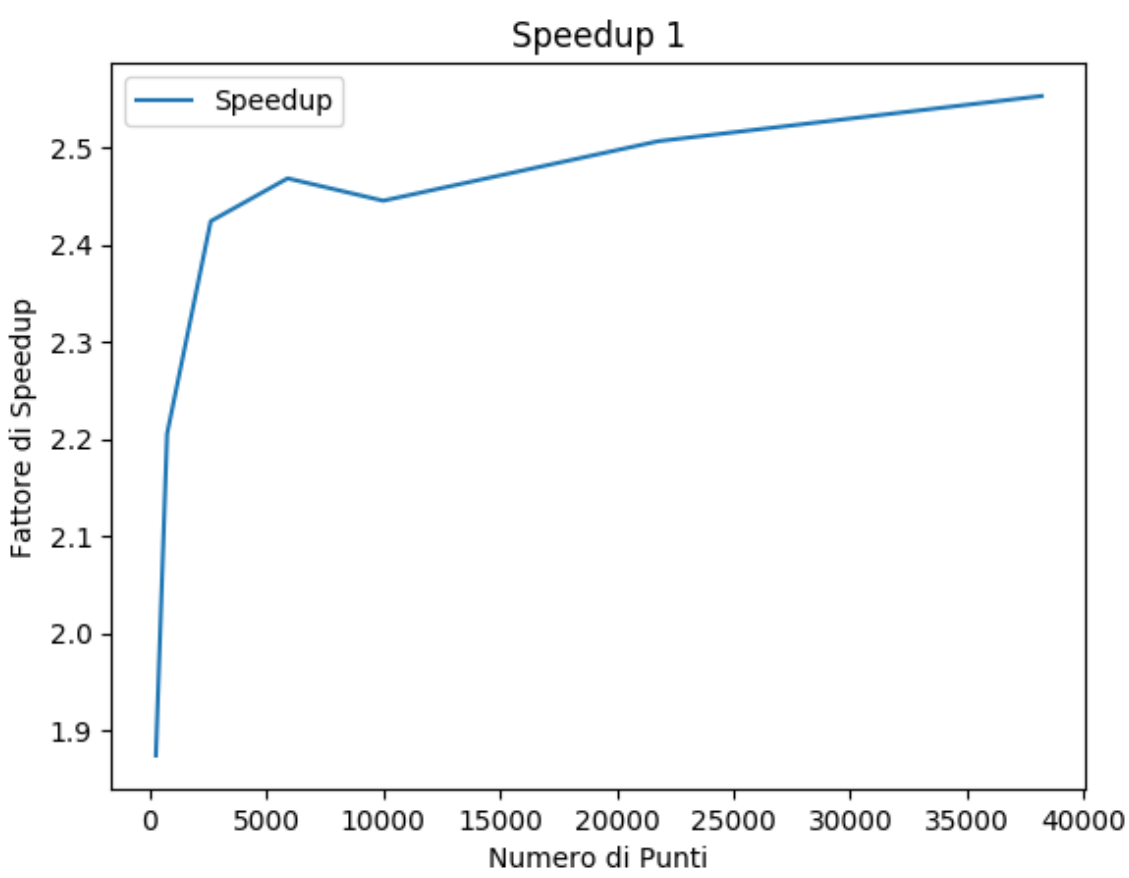

Fig.1: Speedup del calcolo parallelo al variare del numero di punti.

Lo speedup aumenta in maniera simil logaritmica all'aumentare del numero di punti: prima cresce abbstanza rapidamente passando da un numero esiguo di punti a qualche migliaia, poi si assesta e cresce lentamente.

Lo speedup massimo visibile nel grafico è poco superiore a 2.5, dunque lineare (dato che i core a disposizione sono 8). In effetti, la *slackness* **per kmeans parallelo è pari a** *(n/log(n))/P* **: nel nostro caso n(massimo) = 38183 e P = 8, dunque la slackness è ~ 313, un numero molto elevato che fa capire che il numero di core a disposizione limita fortemente il raggiungimento di uno speedup lineare perfetto.**

Domanda 2

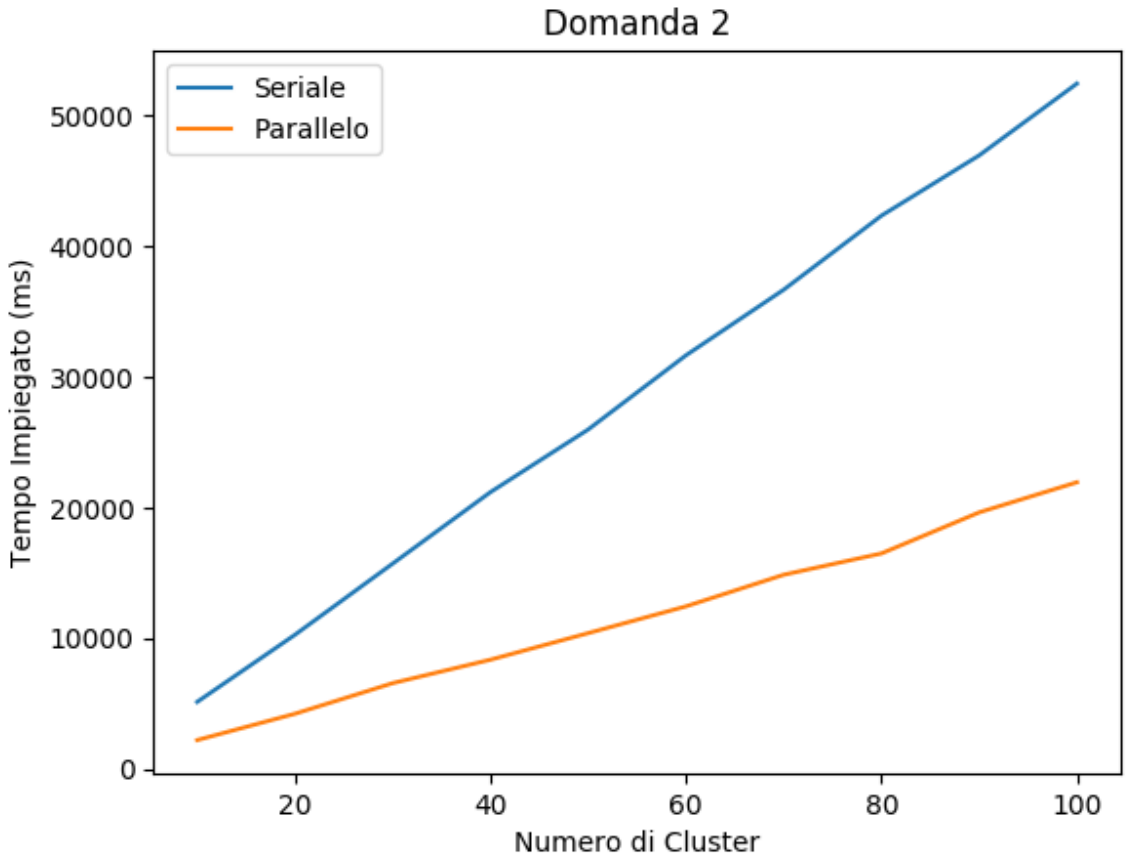

Fig.2: Confronto tempi di calcolo seriale e parallelo al variare del numero di cluster, con numero di punti costante a 38183 e numero di iterazioni costante a 100; si nota anche in questo caso come la versione parallela sia di gran lunga più veloce della versione seriale: entrambe le rette (al netto dei piccoli flessi) crescono in maniera lineare all'aumentare del numero di cluster, ma quella relativa al parallelo ha una pendenza visivamente minore.

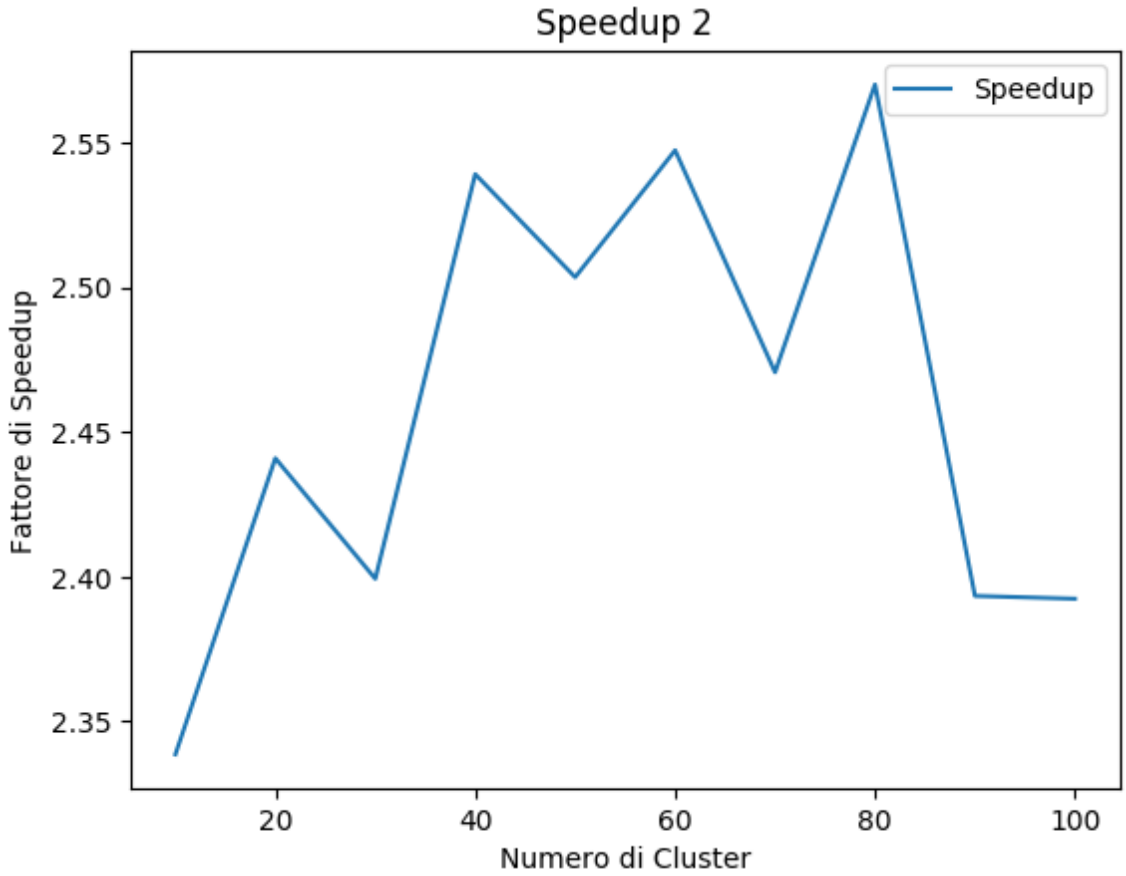

Fig.1: Speedup calcolo parallelo al variare del numero di cluster.

Lo speedup non varia all'aumentare del numero di cluster, ma rimane in un intervallo abbastanza definito (2.30 - 2.60).

Domanda 3

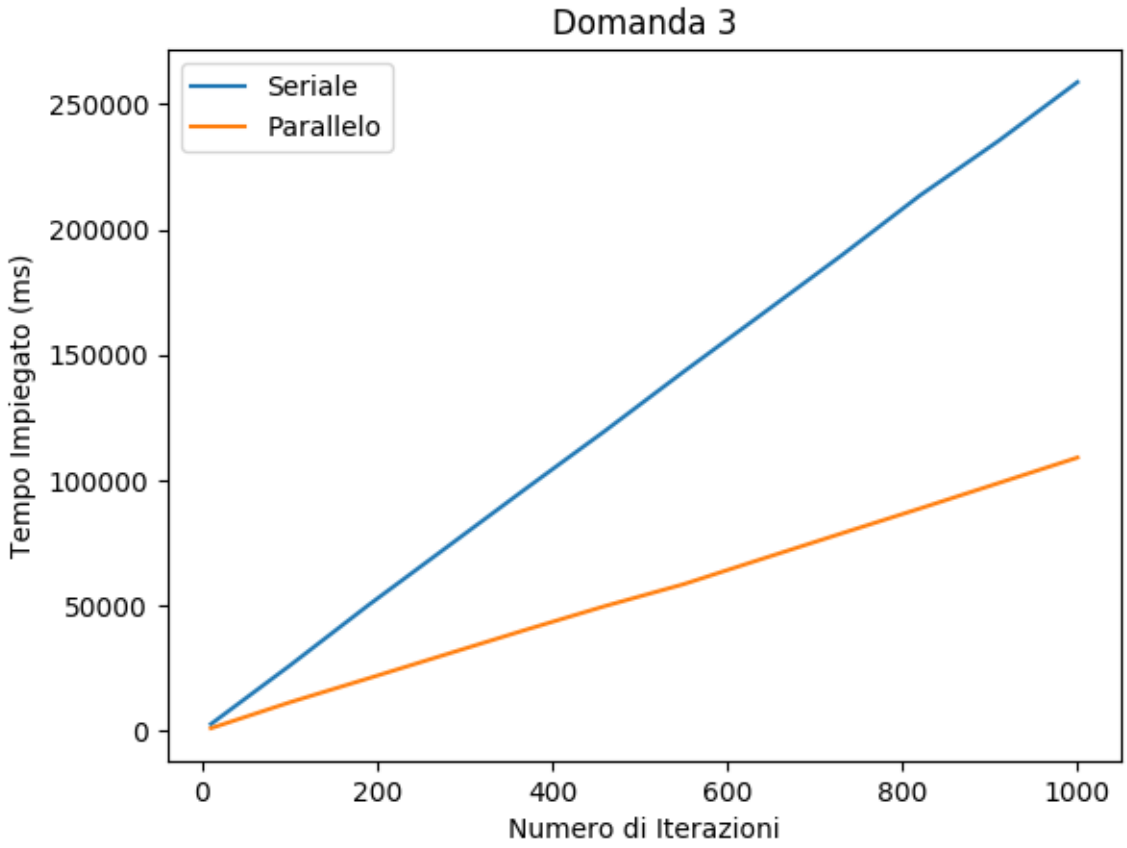

Fig.1: Confronto tempi di calcolo seriale e parallelo al variare del numero di iterazioni, con numero di cluster costante a 50 e numero di punti costante a 38183; si nota ancora come la versione parallela sia di gran lunga più veloce della versione seriale: entrambe le rette crescono in maniera lineare, ma quella relativa al parallelo ha una pendenza visivamente minore

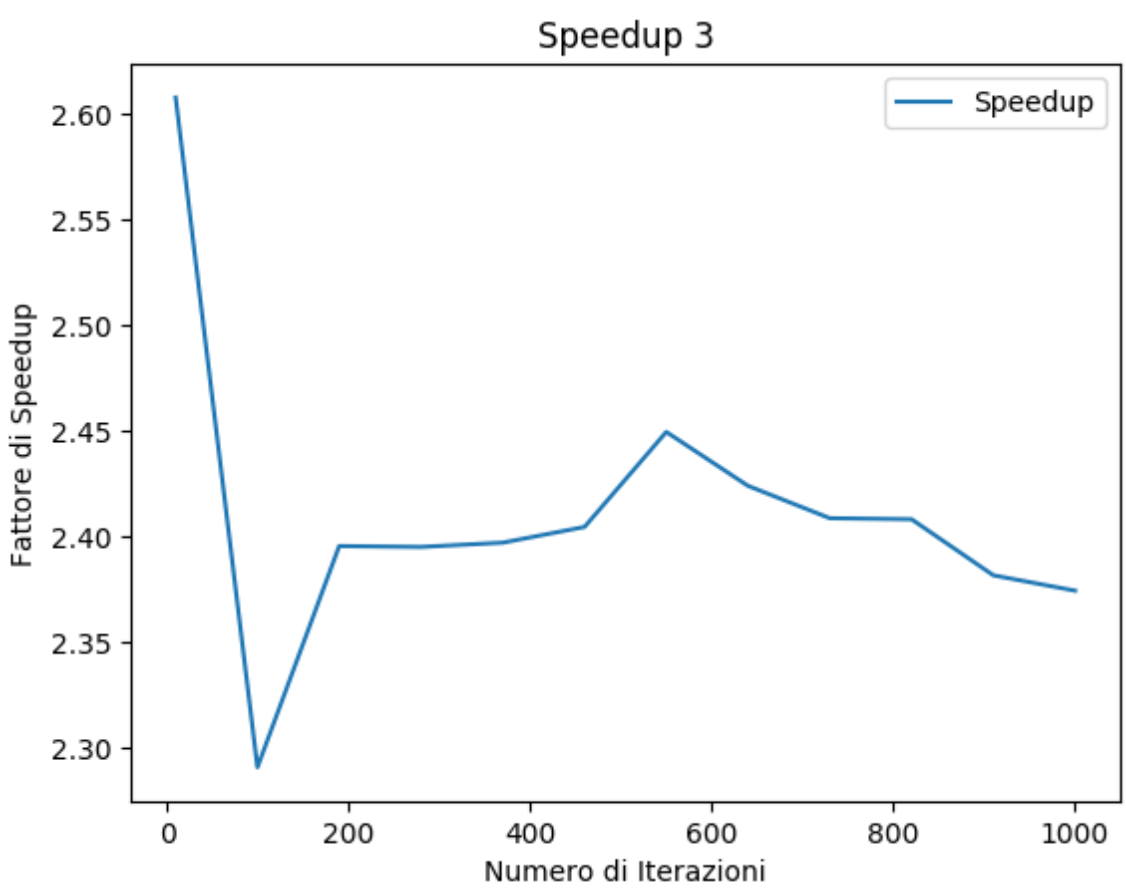

Fig.1: Speedup calcolo parallelo al variare del numero di iterazioni.

Lo speedup sembra essere costante rispetto al numero di iterazioni, infatti il fattore rimane sempre tra 2,5 e 2, questo perché ogni iterazione dell'algoritmo parallelo fa chiamate parallele mentre ogni iterazione dell'algoritmo seriale fa le stesse chiamate ma in versione seriale, quindi la differenza tra i tempi di esecuzione aumenta in maniera lineare rispetto al numero di iterazioni.

Domanda 4

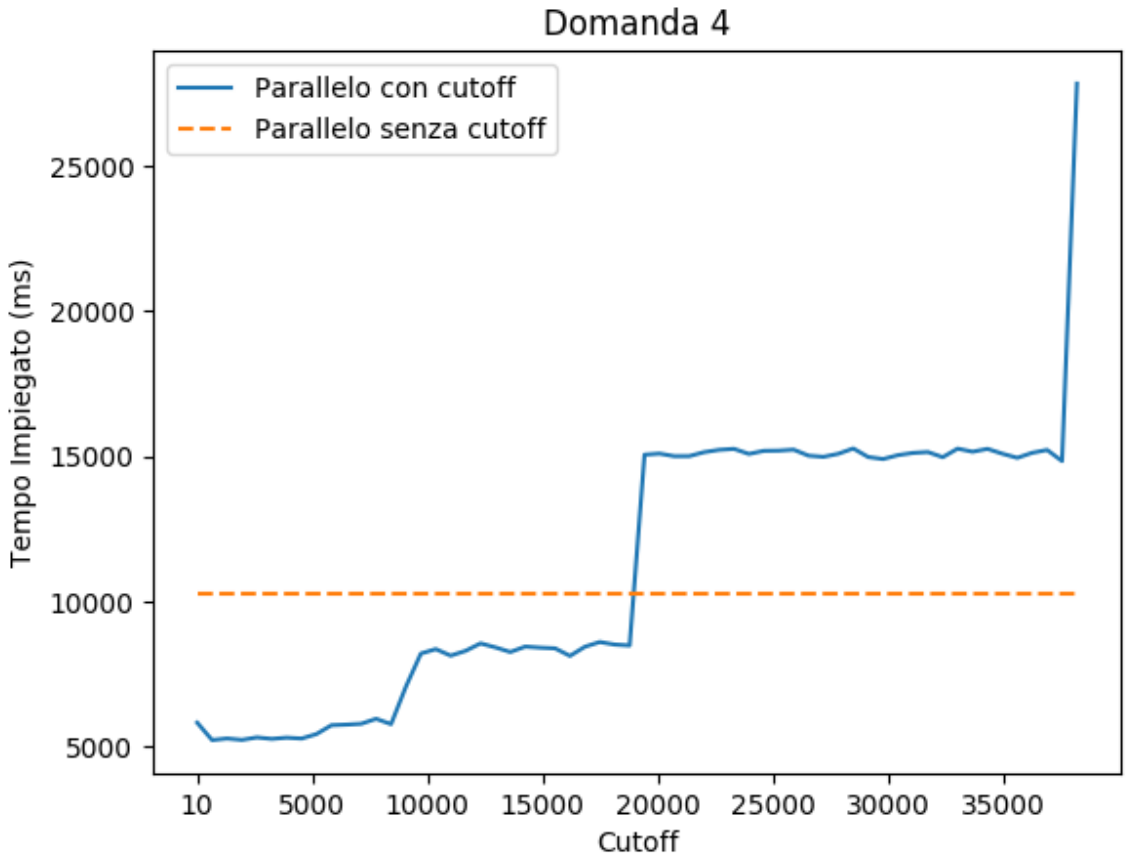

Fig.1: Confronto tempi di calcolo parallelo al variare della soglia di cutoff, con numero di cluster costante a 50 e numero di iterazioni costante a 100.

Il valore di cutoff che ci ha permesso di ottenere le prestazioni migliori è circa 5000: in generale vediamo che un valore di cutoff porta beneficio se non è troppo elevato, altrimenti il parallelismo non viene sfruttato a dovere; in particolare, osservando il valore della retta tratteggiata arancione (che indica il tempo d'esecuzione dell'algoritmo senza cutoff), notiamo che una soglia di cutoff minore della metà dei nodi ci permette di avere un miglioramento in termini di prestazioni: ciò è giustificato dal fatto che quando la versione parallela viene eseguita su un numero di nodi abbastanza basso, i costi aggiuntivi dati dallo scheduling dei processi paralleli eguagliano o superano i tempi effettivi di esecuzione dell'algoritmo.

Domanda 5

Caratteristiche hardware

Processore	Intel® Core™ i7-4710HQ
Numero di core	4 fisici * 2 virtuali
Frequenza base	2.5GHz
Frequenza massima	3.5GHz
Cache	6MB