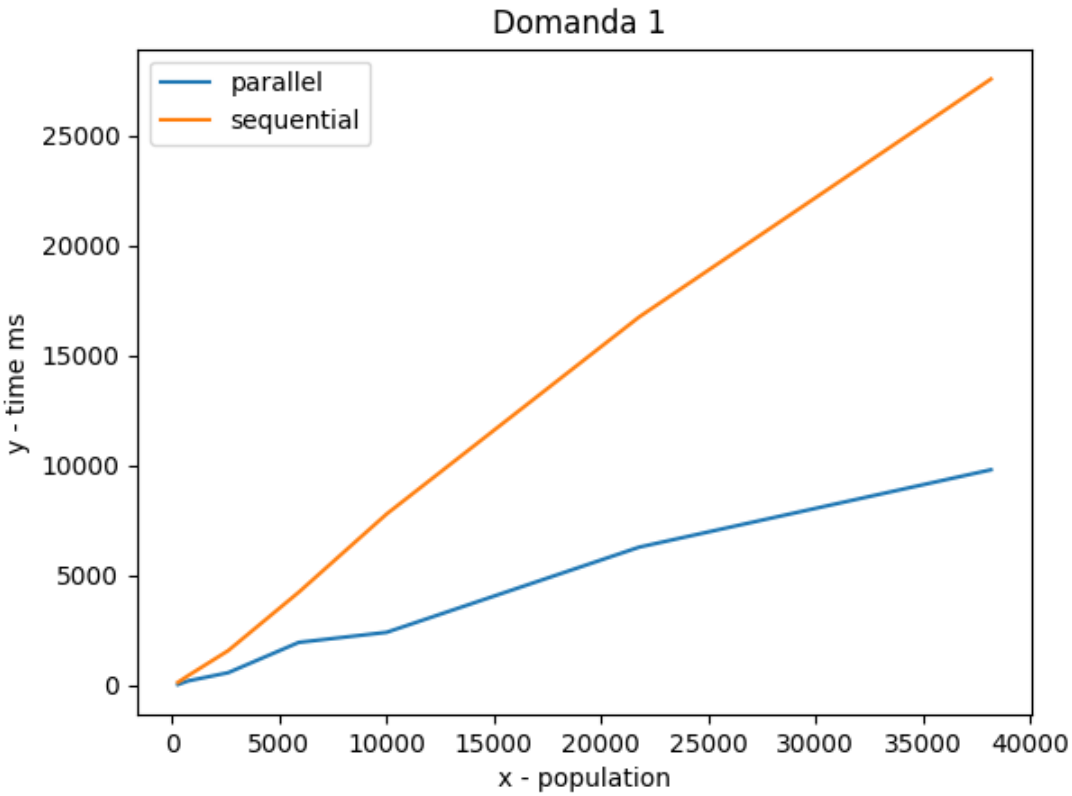
# Lab. 5 - Clustering k-means parallelo

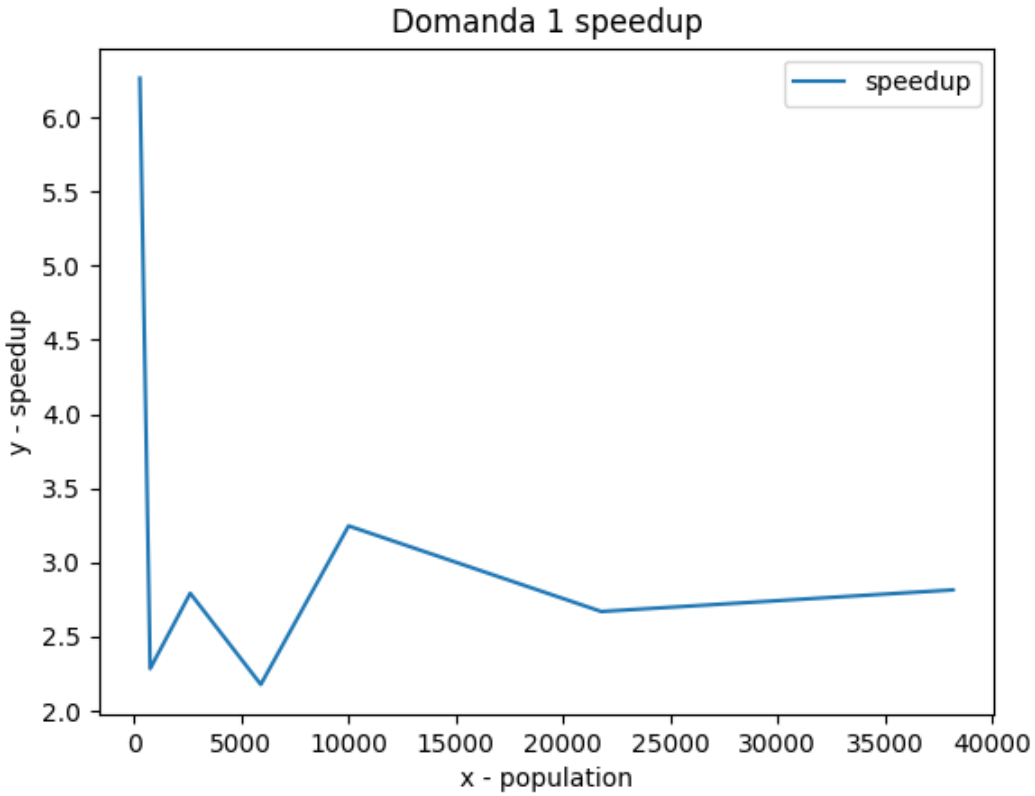
Ballarin Simone, Gobbo Alessio, Rossi Daniel

July 2019

## Domanda 1

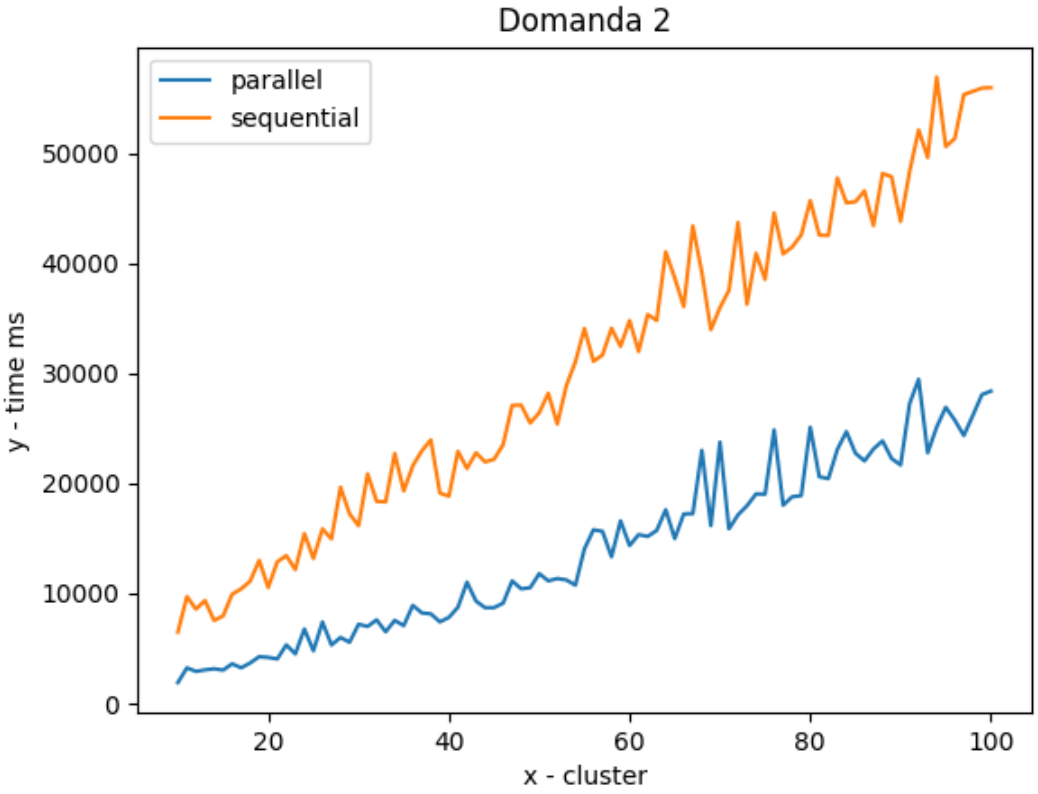



Nel grafico vengono messi a confronto i tempi ottenuti dall'algormitmo k-means, nella sua versione parallela e sequenziale, al variare della popolazione tenendo costanti i parametri: 50 cluster, 100 iterazioni e cutoff 20.
Si è scelto di usare tale cutoff in quanto la versione puramente parallela (cutoff = 0) sarebbe stata penalizzata da un'onere eccessivo di runtime.
Come si evince dal grafico la versione parallela ottiene tempi sempre migliori rispetto alla controparte sequenziale.

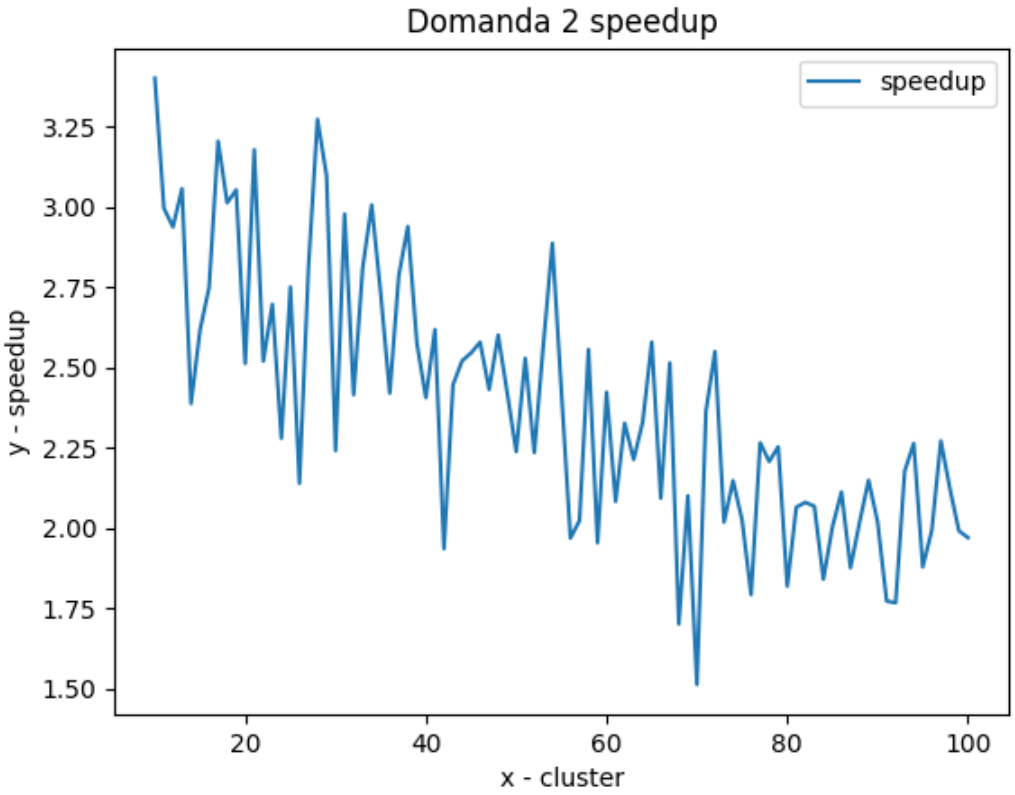



Il grafico mostra lo speedup calcolato come il rapporto tra i tempi dell'algoritmo k-means sequenziale e parallelo, al variare della popolazione.
Dai dati a disposizione sembra evincersi una tendenza positiva in quanto lo *speedup* $> 1$, inoltre si vede che all'aumentare della popolazione aumenta esso stesso.

## Domanda 2

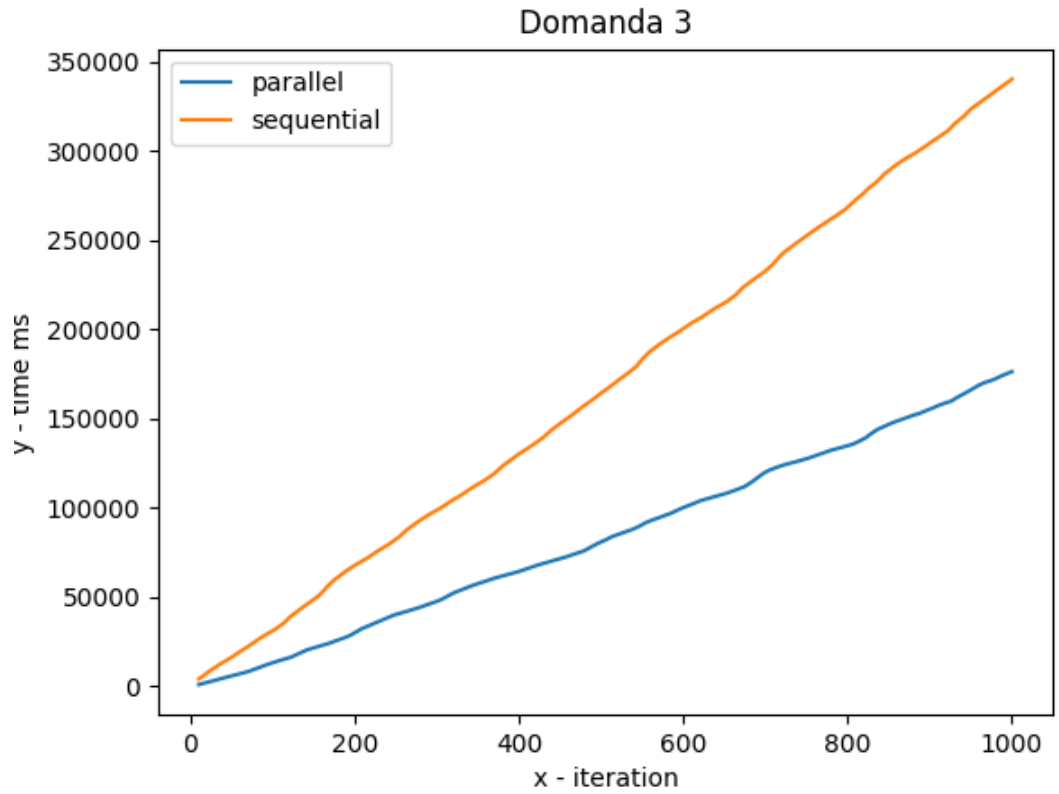



Nel grafico vengono mostrati i tempi di esecuzione dell'algoritmo k-means parallelo e sequenziale, al variare del numero di cluster, calcolato sul dataset complessivo di 38183 punti, con 100 iterazioni e cutoff 20, si può vedere come l'algoritmo parallelo abbia tempi sempre migliori rispetto la controparte sequenziale.

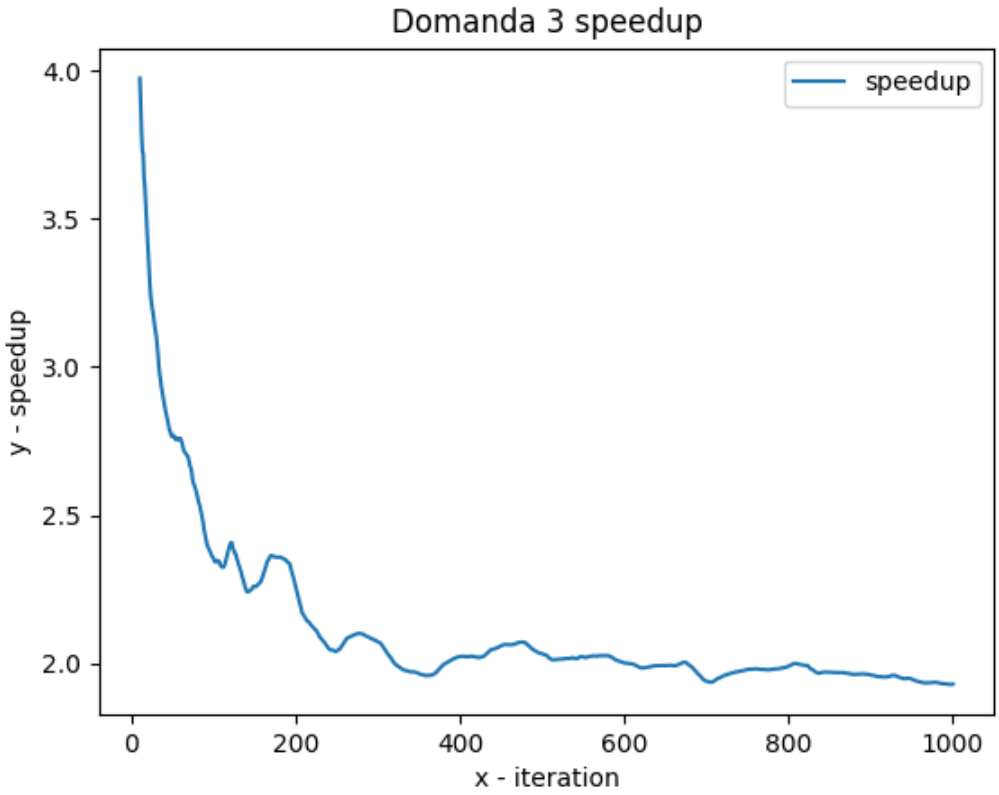



Il grafico mostra l'andamento dello speedup calcolato attraverso il rapporto tra i tempi dell'algoritmo k-means sequenziale e parallelo, al variare del numero di cluster.
Si evince come lo speedup abbia un andamento descrescente, all'aumentare del numero dei cluster, questo perché se aumenta il numero di cluster aumentano le ricerche in parallelo su tutti i punti per calcolare il nuovo centroide, cosa che invece non avviene nella versione sequenziale, in quanto i punti sono già divisi rispetto i cluster.

## Domanda 3

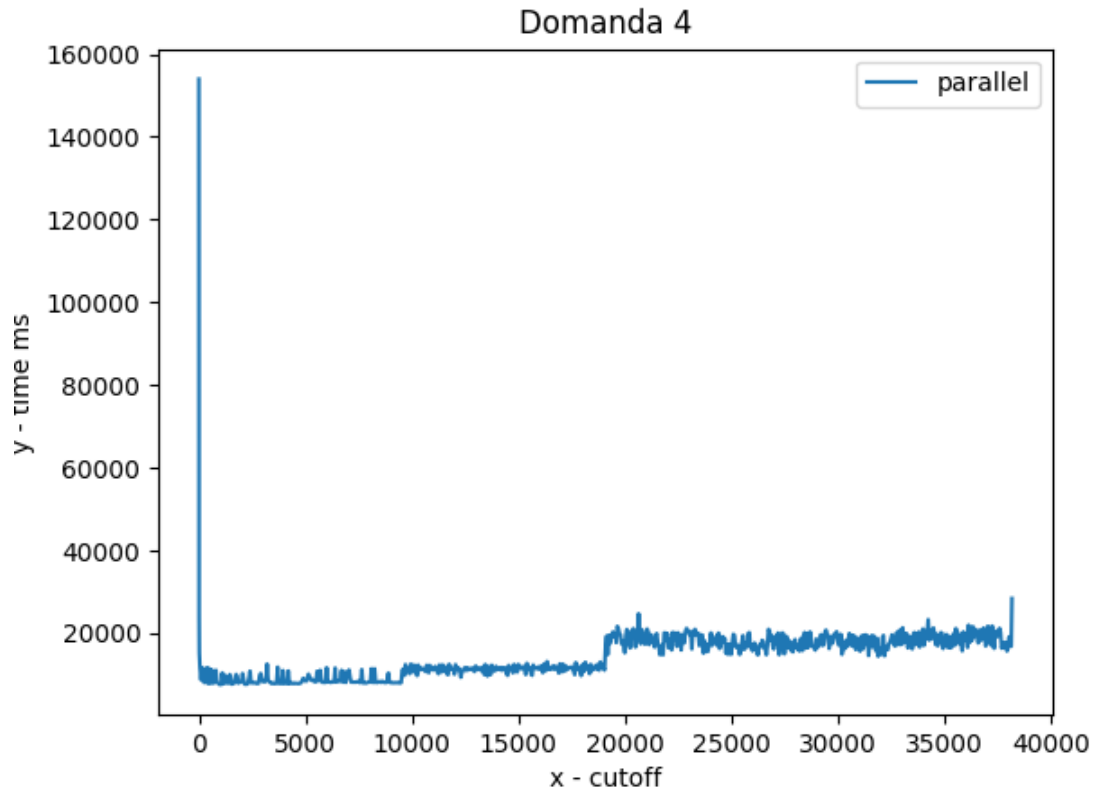



Nel grafico vengono mostrati i tempi di esecuzione dell'algoritmo k-means parallelo e sequenziale, al variare del numero di iterazioni, calcolato sul dataset complessivo di 38183 punti, 50 cluster e cutoff 20, si può vedere come l'algoritmo parallelo abbia tempi sempre migliori rispetto la controparte sequenziale.




Il grafico mostra l'andamento dello speedup calcolato attraverso il rapporto tra i tempi dell'algoritmo k-means sequenziale e parallelo, al variare del numero di iterazioni. Si evince come lo speedup abbia un andamento descrescente, all'aumentare del numero dei iterazioni, attestandosi nella parte finale (iterazioni $\approx 1000$) a circa 2.

## Domanda 4




Nel grafico vengono mostrati i tempi di esecuzione dell'algoritmo k-means parallelo, al variare del cutoff, calcolato sul dataset complessivo di 38183 punti, 50 cluster, 100 iterazioni e con un passo tra un cutoff e il successivo pari a 50.

| $min_{cutoff}$ | $max_{cutoff}$ | Thread |
|---|---|---|
| 38183 | 38183 | 1 |
| 19091 | 38182 | 2 |
| 9545 | 190910 | 4 |
| 4772 | 9544 | 8 |
| 2386 | 4771 | 16 |
| ... | ... | ... |
| 0 | 0 | 38183 |

Come sottolineato dall'andamento a gradoni e facendo riferimento alla tabella riportata qui di fianco, si può vedere come per intervalli di cutoff via via sempre più piccoli rispetto al massimo rappresentante di cutoff disponibile, il tempo di esecuzione tende a rudirsi.
Trovandosi ad avere cutoff molto bassi si può dire di essere in prossimità di un parallelismo puro (cutoff $\approx 0$) mentre per cutoff molto alti ci si trova in prossimità della versione della versione sequenziale dell'algoritmo (cutoff $\approx$ 38183).
Possiamo notare come per cutoff appartenenti allo stesso intervallo si abbiano tempi di esecuzione simili.

L'algoritmo parallelo da il meglio di sè utilizzando un cutoff compreso nell'intervallo $[500, 5000]$, ne troppo basso (parallelismo perfetto), ne troppo alto (praticamente sequenziale), favorendo quindi una versione ibrida.

# Domanda 5

Il computer su cui sono stati eseguiti i test dispone del seguente processore: *Intel(R) Core(TM) i7-2820QM CPU*
*Clockspeed: 2.3 GHz, Turbo Speed: 3.4 GHz*
*Number of Cores: 4 (2 logical cores per physical).*

# Domanda 6

Per sviluppare le domande della relazione abbiamo creato un file *istance.csv* al cui interno abbiamo inserito per ogni riga un'istanza da eseguire nel seguente formato:

**dom,pop,k,it,cutoff**

Per lanciare l'esecuzione si deve eseguire il seguente comando *go run ./main.go ./kmeans.go ./parser.go.*
I risultati verranno inseriti nello stesso ordine di quello dell'input all'interno del file *result.csv* nel seguente formato:

**dom pop k it cutoff tempo_par tempo_seq**